
Trusted and Privacy-preserving Embedded Systems

Advances in Design, Analysis and Application of Lightweight Privacy-preserving Authentication and Physical Security Primitives

Vom Fachbereich Informatik (FB 20)
an der Technischen Universität Darmstadt
zur Erlangung des akademischen Grades eines Doktor-Ingenieurs
genehmigte Dissertation von:

Dipl.-Ing. Christian Wachsmann
Geboren am 26. April 1981 in Coburg, Deutschland

Referenten

Prof. Dr.-Ing. Ahmad-Reza Sadeghi (Erstreferent)
Prof. Dr. Ir. Bart Preneel (Zweitreferent)

Tag der Einreichung: 29. Juli 2013
Tag der Disputation: 25. September 2013



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Darmstadt, 2014
Hochschulkennziffer: D17

Zusammenfassung

RFID (Radio Frequency Identification) Systeme ermöglichen RFID-Lesegeräten die automatische drahtlose Identifikation von RFID-Tags und stellen eine allgegenwärtige Technologie mit zahlreichen Anwendungsmöglichkeiten dar, darunter Zugangskontrollsysteme, elektronische Tickets, Zahlungssysteme [9, 178, 141, 177] und elektronische Ausweisdokumente [93]. Neben ihren Vorteilen bringen RFID-Systeme viele herausfordernde Risiken mit sich [196, 96], insbesondere hinsichtlich des Daten- und Privatsphäreschutzes ihrer Nutzer. Der unsachgemäße Einsatz von RFID kann sensitive Informationen über Anwender und deren Aufenthaltsorte preisgeben und die Erstellung von Nutzerprofilen ermöglichen. Zusätzlich müssen RFID-Systeme die klassischen Sicherheitsziele von Authentifikations- und Identifikationssystemen erfüllen und sicher gegen Angreifer sein, die versuchen sich unberechtigt als legitimes RFID-Tag auszugeben (Impersonierung) oder ein solches zu kopieren (klonen). Für den praktischen Einsatz von RFID ist es daher unerlässlich Sicherheits- und Datenschutzanforderungen zu ermitteln und durchzusetzen.

Diese Arbeit gibt einen umfassenden Überblick über den Stand der Technik sicherer und privatsphäreschützender RFID-Systeme, insbesondere Lösungen basierend auf Physically Unclonable Functions (PUFs). Sie präsentiert Forschungsergebnisse und Fortschritte beim Design, der Analyse und der Evaluation von sicheren und privatsphäreschützenden Authentifikationsmechanismen für RFID-Systeme und PUFs.

Für die Entwicklung von beweisbar sicheren und privatsphäreschützenden RFID-Protokollen ist es unabdingbar die zu erfüllenden Sicherheits- und Datenschutzanforderungen formal zu beschreiben. Jedoch sind bestehende Sicherheits- und Datenschutzmodelle für RFID-Systeme [143, 95, 10, 11, 45, 99, 190, 46, 78, 137, 35] sehr unterschiedlich und bilden oftmals nicht alle Möglichkeiten realer Angreifer ab. Diese Arbeit untersucht die Modellierung von Sicherheits- und Datenschutzaspekten in einem der umfassendsten RFID-Sicherheits- und Datenschutzmodelle [190, 150], welches als Grundlage vieler weiterer Arbeiten dient [137, 138, 32, 165, 164, 50, 49, 168, 167]. Die Arbeit zeigt, dass subtile Aspekte, wie die Modellierung der Kompromittierung von Tags, zur Unerreichbarkeit von gegenseitiger Authentifikation bei gleichzeitigem Schutz der Privatsphäre führen können.

Diese Ergebnisse führten zur Verbesserung des betrachteten RFID-Modells [191] und wurden in Folgearbeiten zu privatsphäreschützenden RFID-Systemen berücksichtigt [84, 53].

Ein vielversprechender Ansatz den Schutz der Privatsphäre in RFID Systemen zu verbessern ohne die Anforderungen an die zugrundeliegenden RFID-Tags zu erhöhen, sind *Anonymizer* [97, 74, 169, 7]. Dies sind spezielle Geräte, die den RFID-Tags die für privatsphäreschützende Protokolle typischen, aufwendigen Berechnungen abnehmen. Während bestehende auf Anonymizern basierende Verfahren anfällig für Impersonierungs- und Denial-of-Service-Angriffe sind, werden Anonymizer in bestehenden RFID-Sicherheitsmodellen nicht betrachtet. Diese Arbeit stellt das erste Sicherheitsmodell für RFID-Systeme mit Anonymizern vor und präsentiert zwei privatsphäreschützende Authentifikationsprotokolle für RFID-Systeme, die Anonymizer verwenden. Beide Verfahren haben attraktive Eigenschaften, die von bisherigen Lösungen nicht gleichzeitig erreicht wurden. Das erste Protokoll ist hocheffizient für alle beteiligten Parteien, schützt die Privatsphäre der Anwender auch dann, wenn RFID-Tags korrumpiert wurden, und ist sicher gegen Impersonierungs-Angriffe und Fälschungen von RFID-Tags selbst wenn der Angreifer die Anonymizer korrumpieren kann. Das zweite Verfahren bietet erstmalig Anonymität und Unverfolgbarkeit von RFID-Tags, die keine Public-Key Operationen (d. h. modulare Exponentiationen) durchführen können, gegenüber Lesegeräten sowie sichere Tag-Authentifikation gegen kollaborierende bössartige Lesegeräte und Anonymizer.

Die in der Praxis üblicherweise eingesetzten RFID-Tags sind kosteneffiziente Geräte ohne teure Schutzmechanismen [9, 141, 177]. Physically Unclonable Functions (PUFs) [67, 156, 184, 87, 24, 54, 148, 167, 192] sind ein vielversprechender Ansatz RFID-Tags effizient vor Hardwareangriffen [140, 43, 104] zu schützen. Jedoch sind bestehende PUF-basierte RFID-Authentifikationsmechanismen nicht skalierbar, erlauben nur eine begrenzte Anzahl von Authentifikationen und sind anfällig für Replay-, Denial-of-Service- und Emulationsangriffe. Diese Arbeit präsentiert zwei skalierbare PUF-basierte Authentifikationsverfahren, die diese Probleme nicht haben. Das erste Verfahren ermöglicht die gegenseitige Authentifikation von RFID-Tag und Lesegerät, bietet Schutz vor Emulationsangriffen und ist äußerst skalierbar. Das zweite Protokoll verwendet einen PUF-basierten Schlüsselspeicher und adressiert eine offene Forschungsfrage zur Realisierbarkeit eines Authentifikationsprotokolls mit der *destructive-private* Eigenschaft [190], d. h., eines privatsphäreschützenden Authentifikationsverfahren bei dem das Kompromittieren der RFID-Tags immer zu deren Zerstörung führt.

Die Sicherheit von PUFs basiert auf Annahmen über physikalische Eigenschaften. Bekannte Evaluierungsergebnisse von PUFs basieren auf unterschiedlichen Analysemethoden und verschiedenen Testbedingungen und sind daher nur schwer vergleichbar. Diese Arbeit präsentiert die erste umfangreiche Analyse der fünf bekanntesten PUF-Implementierungen in ASIC, darunter Arbiter, Ring Oscillator, SRAM, Flip-Flop und Latch PUFs. Die Arbeit stellt ein neues Evaluierungsverfahren für PUFs vor und quantifiziert die wichtigsten Eigenschaften von PUFs für deren Verwendung in kryptographischen Verfahren.

PUFs wurden für eine Vielzahl von Anwendungen vorgeschlagen, darunter auch Authentifikationsmechanismen und Mechanismen zum Schutz vor Produktfälschungen. Jedoch gibt es bisher nur rudimentäre Sicherheitsmodelle für PUFs, was das Vertrauen in die Sicherheitseigenschaften bestehender PUF-basierter Sicherheitsmechanismen stark begrenzt. Diese Arbeit präsentiert das erste formale Sicherheitsmodell für PUF-basierte Primitiven. Dieses Modell wurde bisher dazu verwendet die Eigenschaften von bildbasierten PUFs zu modellieren [174] und beim Design von Mechanismen zum Schutz vor Produktfälschungen [173] sowie von physikalischen Hash-Funktionen [58] eingesetzt.

Summary

Radio Frequency Identification (RFID) enables RFID *readers* to perform fully automatic wireless identification of objects labeled with RFID *tags* and is widely deployed to many applications, such as access control, electronic tickets and payment [9, 178, 141, 177] as well as electronic passports [93]. This prevalence of RFID technology introduces various risks [196, 96], in particular concerning the privacy of its users and holders. Despite the privacy risk, classical threats to authentication and identification systems must be considered to prevent the adversary from impersonating or copying (cloning) a tag.

This thesis summarizes the state of the art in secure and privacy-preserving authentication for RFID tags with a particular focus on solutions based on Physically Unclonable Functions (PUFs). It presents advancements in the design, analysis and evaluation of secure and privacy-preserving authentication protocols for RFID systems and PUFs.

Formalizing the security and privacy requirements on RFID systems is essential for the design of provably secure and privacy-preserving RFID protocols. However, existing RFID security and privacy models in the literature [143, 95, 10, 11, 45, 99, 190, 46, 78, 137, 35] are often incomparable and in part do not reflect the capabilities of real-world adversaries. We investigate subtle issues such as tag corruption aspects that lead to the impossibility of achieving both mutual authentication and any reasonable notion of privacy in one of the most comprehensive security and privacy models [190, 150], which is the basis of many subsequent works [137, 138, 32, 165, 164, 50, 49, 168, 167]. Our results led to the refinement of this privacy model [191] and were considered in subsequent works on privacy-preserving RFID systems [84, 53].

A promising approach to enhance the privacy in RFID systems without lifting the computational requirements on the tags are *anonymizers* [97, 74, 169, 7]. These are special devices that take off the computational workload from the tags. While existing anonymizer-based protocols are subject to impersonation and denial-of-service attacks, existing RFID security and privacy models do not include anonymizers. We present the first security and privacy framework for anonymizer-enabled RFID systems and two privacy-preserving RFID authentication schemes using anonymizers. Both schemes

achieve several appealing features that were not simultaneously achieved by any previous proposal. The first protocol is very efficient for all involved entities, achieves privacy under tag corruption. It is secure against impersonation attacks and forgeries even if the adversary can corrupt the anonymizers. The second scheme provides for the first time anonymity and untraceability of tags against readers as well as secure tag authentication against collisions of malicious readers and anonymizers using tags that cannot perform public-key cryptography (i.e., modular exponentiations).

The RFID tags commonly used in practice are cost-efficient tokens without expensive hardware protection mechanisms [9, 141, 177]. Physically Unclonable Functions (PUFs) promise to provide an effective security mechanism for RFID tags [67, 156, 184, 87, 24, 54, 148, 167, 192] to protect against basic hardware attacks [140, 43, 104]. However, existing PUF-based RFID authentication schemes are not scalable, allow only for a limited number of authentications and are subject to replay, denial-of-service and emulation attacks. We present two scalable PUF-based authentication schemes that overcome these problems. The first protocol supports tag and reader authentication, is resistant to emulation attacks and highly scalable. The second protocol uses a PUF-based key storage and addresses an open question on the feasibility of *destructive privacy* [190], i.e., the privacy of tags that are destroyed during tag corruption.

The security of PUFs relies on assumptions on physical properties and is still under investigation. PUF evaluation results in the literature are difficult to compare due to varying test conditions and different analysis methods. We present the first large-scale security analysis of ASIC implementations of the five most popular electronic PUF types, including Arbiter, Ring Oscillator, SRAM, Flip-Flop and Latch PUFs. We present a new PUF evaluation methodology that allows a more precise assessment of the unpredictability properties than previous approaches and we quantify the most important properties of PUFs for their use in cryptographic schemes.

PUFs have been proposed for various applications, including anti-counterfeiting and authentication schemes. However, only rudimentary PUF security models exist, limiting the confidence in the security claims of PUF-based security mechanisms. We present a formal security framework for PUF-based primitives, which has been used in subsequent works to capture the properties of image-based PUFs [174] and in the design of anti-counterfeiting mechanisms [173] and physical hash functions [58].

Acknowledgements

First and foremost I thank my thesis advisor Prof. Ahmad-Reza Sadeghi for his excellent supervision and continuous support. I am very honoured to have Prof. Bart Preneel as thesis co-advisor and thank him for his valuable feedback. Moreover, I thank Prof. Johannes Buchmann, Prof. Max Mühlhäuser and Prof. Michael Waidner for being members of my committee.

Special thanks go to Prof. Ivan Visconti and Prof. Frederik Armknecht for the countless inspiring discussions. I always enjoyed working with you. Further, I thank all my co-authors beyond Ahmad, Ivan and Frederik with whom I collaborated during my doctoral studies (Christoph Busold, Liqun Chen, Kurt Dietrich, Alexandra Dmitrienko, Anthony van Herrewege, Prof. Stefan Katzenbeisser, Ünal Koçabas, Vincent van der Leest, Hans Löhr, Roel Maes, Yossef Oren, Roel Peeters, Vladimir Rožić, Alessandra Scafuro, Steffen Schulz, Hervé Seudié, Geert-Jan Schrijen, Heike Schröder, Majid Sobhani, Prof. François-Xavier Standaert, Ahmed Taha, Sandeep Tamrakar, Prof. Ingrid Verbauwhede, and Johannes Winter).

My research work and travels were in part funded by the European Commission through the projects SPEED, ECRYPT and UNIQUE. These projects provided an excellent working environment and allowed me to collaborate with renowned and leading researchers from academia and industry. In particular, I thank Intel, Intrinsic ID, KU Leuven and Sirrix for the development of the ASIC and evaluation board used in the security analysis of the implementations of Physically Unclonable Functions (PUFs), which is essential part of my thesis. Further, I thank Patrick Koeberl, Prof. Pim Tuyls and Jérôme Quevremont for many valuable discussions on PUF designs and hardware attacks as well as Timm Korte for providing his implementation of PRESENT.

I thank all my colleagues at the System Security Labs at Ruhr-Universität Bochum and Technische Universität Darmstadt for many fruitful and inspiring technical discussions, their help in organizational matters and the great time we had.

Additionally, I thank my parents and my family for their continuous encouragement. Last but not least, a special thank goes to Julia for her enduring support and sympathy.

Contents

Summary	v
Acknowledgements	ix
Contents	xiv
1 Introduction	3
1.1 Summary of Main Results	5
1.2 Outline	7
2 Preliminaries	9
2.1 Notation	9
2.2 Mathematic and Cryptographic Background	9
2.2.1 Bilinear Maps	9
2.2.2 Intractability Assumptions	10
2.3 Cryptographic Primitives	11
2.3.1 Pseudo-random Functions	11
2.3.2 Random Oracles	11
2.3.3 Hash Functions	12
2.3.4 Encryption Schemes	12
3 Background on RFID	15
3.1 System Architecture and Requirements	16
3.1.1 Common System Architecture	16
3.1.2 Attacks on RFID Systems	17
3.1.3 Trust and Adversary Models	18
3.1.4 Requirement Analysis	20
3.2 Existing Security and Privacy Models	22
3.3 Analysis of Existing Solutions	24
3.3.1 Physical Methods	24
3.3.2 Anonymous Authentication Protocols	25
3.3.3 Privacy-preserving Authentication Protocols	25
3.4 Conclusion and Open Problems	31
4 RFID Security and Privacy Revisited	33
4.1 Motivation and Contribution	33

4.2	The Paise-Vaudenay RFID Security and Privacy Model	36
4.2.1	System Model	36
4.2.2	Trust and Adversary Model	37
4.2.3	Security Definition	39
4.2.4	Privacy Definition	40
4.3	The Paise-Vaudenay Model Revisited	42
4.3.1	Corruption with Full State Disclosure	42
4.3.2	Corruption without Temporary State Disclosure	46
4.4	Conclusion	49
5	Anonymizer-enabled Security and Privacy for RFID	51
5.1	Motivation and Contribution	51
5.2	Anonymizer-enabled RFID Systems	54
5.3	Anonymizers in RFID Systems with Trusted Readers	55
5.3.1	Trust Model and Assumptions	56
5.3.2	Protocol Specification	58
5.3.3	Performance Evaluation	61
5.3.4	Security Analysis	61
5.4	Anonymizers in RFID Systems with Untrusted Readers	72
5.4.1	Trust Model and Assumptions	73
5.4.2	Protocol Specification	74
5.4.3	Performance Evaluation	79
5.4.4	Security Analysis	80
5.5	Conclusion	85
6	Background on Physically Unclonable Functions	87
6.1	PUF Concept, Properties and Assumptions	87
6.2	PUF Types	89
6.3	Noise Compensation and Privacy Amplification	89
6.4	Common PUF-based Applications	90
6.4.1	Device Identification and Authentication	90
6.4.2	Secure Key Generation and Storage	91
6.5	Attacks on PUFs and PUF-based Systems	92
6.5.1	Emulation Attacks	92
6.5.2	Side Channel Attacks	92
6.5.3	Fault Injection Attacks	93
6.6	Advanced PUF Concepts	94
6.6.1	Controlled PUFs	94
6.6.2	Reconfigurable PUFs	94
6.7	Conclusion and Open Problems	96

7	Security Evaluation of PUF Implementations on ASIC	97
7.1	Motivation and Contribution	97
7.2	The PUF ASIC	98
7.3	Evaluation Methodology	100
7.3.1	Robustness Analysis	101
7.3.2	Unpredictability Analysis	102
7.4	Evaluation Results	107
7.4.1	Robustness Results	107
7.4.2	Unpredictability Results	109
7.4.3	Discussion	114
7.4.4	Summary	114
7.5	Conclusion	115
8	Formal Model of Physically Unclonable Functions	117
8.1	Motivation and Contribution	117
8.2	Framework for Physical Functions	119
8.2.1	Background and Rationale	119
8.2.2	Formalization	121
8.3	Robustness	123
8.3.1	Rationale	123
8.3.2	Formalization	123
8.4	Physical Unclonability	125
8.4.1	Rationale	125
8.4.2	Formalization	126
8.5	Unpredictability	129
8.5.1	Rationale	129
8.5.2	Formalization	130
8.6	Conclusion	132
9	PUF-enhanced Security and Privacy for RFID	135
9.1	Motivation and Contribution	135
9.2	Lightweight PUF-based RFID Authentication	136
9.2.1	Reverse Fuzzy Extractors	137
9.2.2	Protocol Description and Specification	138
9.2.3	Performance Evaluation	141
9.2.4	Security Analysis	142
9.2.5	Conclusion	147
9.3	Privacy-preserving PUF-based RFID Authentication	148
9.3.1	Protocol Description and Specification	149
9.3.2	Security Analysis	151
9.3.3	Conclusion	158

10 Application Example: RFID-based E-Tickets	159
10.1 General Scenario	159
10.2 Requirement Analysis	160
10.3 E-Ticket Systems in Practice	161
10.3.1 Calypso E-Ticket Standard	161
10.3.2 Other E-Ticket Systems	162
10.4 Secure and Privacy-preserving Protocols for E-Tickets	163
10.4.1 Existing Solutions	163
10.4.2 Anonymizer-based Solutions	163
10.4.3 PUF-based Solutions	164
11 Conclusion	165
11.1 Summary	165
11.2 Directions for Future Research	167
Bibliography	171
About the Author	189

1 Introduction

Radio Frequency Identification (RFID) enables RFID *readers* to perform fully automatic wireless identification of objects labeled with RFID *tags* and is widely deployed to many applications, such as access control (e.g., to buildings and public transit systems) [9, 178, 141, 177] and electronic passports [93]. As pointed out in previous publications [196, 96], this prevalence of RFID technology introduces various risks, in particular concerning the privacy of its users and holders.

The most deterrent risk of RFID systems are *tracing attacks* that aim at creating user profiles without knowledge and consent of the user. Due to the wireless interface of RFID, user-related information may be leaked unnoticeably through the wireless interface to unauthorized entities, i.e., those that are not trusted by the user. Thus, an important security objective of RFID systems is to prevent unauthorized access to user-related data (*confidentiality*), unauthorized identification of users (*anonymity*) as well as unauthorized tracing of tags by linking their communication (*unlinkability*).

Despite these privacy risks, classical threats to authentication and identification systems must be considered as well. Indeed, potential threats to RFID systems are attacks where the adversary tries to impersonate or clone a legitimate tag. Legitimate means that the tag has been created by an accredited tag issuer. Thus appropriate countermeasures must be provided (*device and/or user authentication* and *unclonability*). However, there are some other risks such as *denial-of-service* attacks where the adversary unnoticeably interacts with the tags and exploits deficiencies of the underlying protocols to permanently disable legitimate tags, which must also be prevented (*availability*).

A general problem of wireless authentication systems and particularly of RFID systems are attacks, where the adversary uses the communication of a valid tag to make an honest reader accept. Since RFID tags respond to any interrogation without requiring user interaction, this may enable the adversary to authenticate as an honest tag to the reader by relaying the protocol messages of an honest tag [81, 62]. There are different variants of this attack [52, 44, 61]: In a *mafia fraud* the adversary interacts with an honest tag and impersonates this tag to an honest reader. A *terrorist fraud* means that

the adversary must collude with a malicious tag to impersonate this tag to an honest reader. In a *distance fraud* the tag claims to be closer to the reader than it actually is. As a countermeasure, *distance bounding* protocols [18] aim to measure the physical distance between tag and reader based on the round-trip-time of protocol messages. A variety of distance bounding protocols have been proposed in the literature [28, 82, 36, 175, 110, 13, 154, 157], which mainly differ in performance and their security objectives.

There is a large body of literature [143, 95, 10, 11, 45, 99, 190, 46, 78, 137, 35] that aims at formalizing the security and privacy requirements of RFID systems and building systems to achieve these requirements. However, existing RFID security and privacy models in the literature are often incomparable and in part do not reflect the capabilities of real-world adversaries. Often subtle issues, such as tag corruption aspects, are not considered appropriately. Hence, setting up a mature RFID security and privacy model that fulfils the sophisticated requirements of real-life applications is a difficult task.

The design of provably-secure privacy-preserving protocols that are applicable to real-world scenarios is very challenging. There is a large body of literature proposing solutions to this problem [196, 83, 135, 144, 55, 118, 183, 176, 97, 74, 169, 7]. However, almost all of them have deficiencies concerning their deployment in real-world applications. For instance, in many protocols the computational effort of a reader to verify a tag depends on the total number of legitimate tags in the system, which is unacceptable for systems with a large number of tags, such as electronic product labels and electronic tickets. Other protocols require the reader to have a permanent online connection to some trusted database, which is inappropriate for systems that require mobile readers, such as ticket inspectors in public transit systems.¹ It is an open problem [190] whether a privacy notion that is meaningful in practice can actually be achieved by using common RFID tags that are often not capable of performing public-key cryptography.² In this context, anonymizer-based RFID systems [97, 74, 169, 7] promise to enable privacy with cost-efficient and resource-constrained tags. However, existing anonymizer-enabled protocols are subject to impersonation and denial-of-service attacks. Furthermore, anonymizers

¹Note that the widespread deployment of mobile communication devices may enable permanent connectivity of mobile readers in the future. However, there are many applications with high availability requirements that must also work without an online connection, in particular when deployed in areas without the necessary communication infrastructure.

²There are RFID tags that can perform public-key cryptography [132, 71, 14, 199, 142]. However, these tags typically take more than 300 ms to authenticate to a reader, which is too slow for most commercial applications such as electronic transit tickets [178].

are not considered by existing RFID security and privacy models in the literature, which makes it hard to analyze anonymizer-based protocols.

Although many effort has been done to secure the communication of tags and readers against tracking on the protocol level, it has been shown [158, 159, 200, 201] that tags can be identified and tracked based on their physical properties. Specifically, tags of different manufacturers typically implement the analogue circuitry of the radio interface in different ways, which may result in different behavior among different devices and enable their classification. Moreover, manufacturing variations affect the analogue components of the radio interface such that individual devices of the same model and the same manufacturer behave differently on the physical communication layer, which may allow to identify them. These tracking possibilities must be addressed at the physical layer and are not considered in this thesis, which focuses on privacy-protecting solutions on the protocol layer.

The RFID tags commonly used in practice do not feature expensive protection mechanisms against hardware attacks [9, 141]. Hence, the authentication secrets of these tags can be read out by performing basic side-channel and invasive attacks. In this context, Physically Unclonable Functions (PUFs) [156, 184, 148, 167] promise to be a cost-effective hardware security primitive that allows binding the authentication secrets to the underlying tag hardware. In contrast to most cryptographic primitives, whose security can be related to well established (albeit unproven) assumptions, the security of PUFs relies on assumptions on physical properties and is still under investigation. Moreover, existing PUF-based authentication schemes are not scalable, subject to denial-of-service attacks and do not provide mutual authentication between the tag and the reader. Hence, the design of a scalable and lightweight PUF-based mutual RFID authentication protocol is a challenging open problem.

1.1 Summary of Main Results

This thesis extends and improves the current state of the art of privacy-protecting RFID systems by formalizing and designing efficient, secure and privacy-preserving authentication protocols that fulfill the sophisticated requirements of real-life RFID applications. More detailed, the contribution of this work is as follows:

Advancing RFID security and privacy models. We analyze one of the most comprehensive RFID security and privacy models, which generalizes and improves many previous works and has been proposed by Paise and Vaudenay [190, 150]. We point out weaknesses and deficiencies in this model and investigate some subtle issues such as tag corruption aspects. Our results led to the refinement of the Paise and Vaudenay RFID security and privacy model [191] and they were considered in the development of improved security and privacy models for RFID systems [84, 53].

First security and privacy framework for anonymizer-enabled RFID. We present the first security and privacy model for anonymizer-enabled RFID systems and two privacy-preserving anonymizer-based RFID authentication schemes. The first protocol achieves a strong notion of privacy (narrow-strong privacy) without requiring tags to perform expensive public-key operations (i.e., modular exponentiations), thus providing a satisfying notion of privacy for cost-efficient tags. The second scheme provides anonymity and untraceability of tags against readers, tag authentication even against collisions of malicious readers and anonymizers and security against denial-of-service attacks on the protocol level.

PUF-enhanced RFID security and privacy. We present two scalable PUF-based mutual authentication schemes that overcome the drawbacks of existing approaches. In particular, the first protocol supports PUF-based mutual authentication between tags and readers, is resistant to attacks that emulate the PUF in software and that supports a virtually unlimited number of authentications without requiring the reader to store a large number of PUF challenge/response pairs. The scheme is based on reverse fuzzy extractors, a new approach to correct noise in PUF responses that allows for extremely lightweight implementations on the token. The second PUF-based authentication protocol uses the PUF as a secure key storage and addresses the open question [190] on the feasibility of destructive privacy, i.e., the privacy of tags that are destroyed during corruption.

First large-scale analysis of PUF implementations in ASIC. We present the first large-scale security analysis of ASIC implementations of the five most popular intrinsic electronic PUF types, including Arbiter, Ring Oscillator, SRAM, Flip-flop and Latch PUFs and assess their applicability to cryptographic and security applications, such as the authentication of RFID tags. Our analysis is based on PUF data obtained at different

operating conditions from 96 ASICs containing multiple PUF instances. In this context, we present an evaluation methodology and quantify the robustness and unpredictability properties of PUFs, which are fundamental for their integration into security mechanisms, such as authentication protocols. Since all PUFs have been implemented in the same ASIC and analyzed with the same evaluation methodology, our results allow for the first time a fair comparison of their properties.

Formal framework for Physically Unclonable Functions. We present a formal foundation for security primitives based on PUFs, focussing on the main properties at the heart of most published works on PUFs: robustness, unclonability and unpredictability. This work allows for a meaningful security analysis of security primitives taking advantage of physical properties, becoming increasingly important in the development of the next generation of secure information systems. Our framework has been used in subsequent works to estimate the robustness and unclonability properties of image-based PUFs [174], the design of anti-counterfeiting mechanisms [173] and physical hash functions [58].

1.2 Outline

We introduce our notation in Chapter 2 and give an overview of the state of the art in security and privacy in RFID systems in Chapter 3. Then we analyze one of the most advanced security and privacy models [190, 150] in Chapter 4. We introduce our security and privacy model for anonymizer-enabled RFID systems in Chapter 5. Next, we give an overview on the state of the art of Physically Unclonable Functions (PUFs) in Chapter 6, focusing on their use in lightweight authentication schemes. We present our PUF evaluation and its results in Chapter 7. Our formal PUF security framework is described in Chapter 8. We present our PUF-based authentication protocols for RFID in Chapter 9. We provide a use case study and requirement analysis of the emerging application of RFID-based electronic transit tickets in Chapter 10. Finally, we conclude in Chapter 11.

2 Preliminaries

2.1 Notation

For a finite set S , $|S|$ denotes the size of S whereas for an integer (or a bit-string) n the term $|n|$ means the bit-length of n . The term $s \xleftarrow{\$} S$ means the assignment of a uniformly sampled element from S to variable s . We denote with $\text{HW}(x)$ the Hamming weight of a bitstring x , i.e., the number of non-zero bits of x . With $\text{HD}(x, y)$ we denote the Hamming distance between two bit strings x and y , i.e., the number of bits that are different in x and y . We denote with \emptyset the empty string.

Let E be some event (e.g., the result of a security experiment), then $\Pr[E]$ denotes the probability that E occurs. Probability $\epsilon(l)$ is called *negligible* if for all polynomials f it holds that $\epsilon(l) \leq 1/f(l)$ for all sufficiently large l . Moreover, probability $1 - \epsilon(l)$ is called *overwhelming* if $\epsilon(l)$ is negligible.

Let A be a probabilistic algorithm. Then $y \leftarrow A(x)$ means that on input x , A assigns its output to the variable y . The term $[A(x)]$ denotes the set of all possible outputs of A on input x that appear with a probability larger than 0. $A_K(x)$ means that the output of A depends on x and some additional parameter K (such as a secret key). The term $\text{Prot}[A : x_A; B : x_B; * : x_{\text{pub}}] \rightarrow [A : y_A; B : y_B]$ denotes an interactive protocol Prot between two algorithms A and B . Hereby, A (resp. B) gets a private input x_A (resp. x_B) and a public input x_{pub} . While A (resp. B) is operating, it can interact with B (resp. A). After the protocol terminates, A (resp. B) returns y_A (resp. y_B).

2.2 Mathematic and Cryptographic Background

2.2.1 Bilinear Maps

Definition 2.1 (Admissible Pairing). *Let $\mathbb{G}_1, \mathbb{G}_2$ and \mathbb{G}_T be three groups of large prime exponent $q \approx 2^l$ for some security parameter $l \in \mathbb{N}$. The groups \mathbb{G}_1 and \mathbb{G}_2 are written additively with the identity element 0. The group \mathbb{G}_T is written multiplicatively with the identity element 1. A pairing is a mapping $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ with the following*

properties:

1. Bilinear, i.e., for all $P, P' \in \mathbb{G}_1$ and all $Q, Q' \in \mathbb{G}_2$ it holds that

$$e(P + P', Q + Q') = e(P, Q) \cdot e(P, Q') \cdot e(P', Q) \cdot e(P', Q').$$

2. Non-degenerate, i.e., for all $P \in \mathbb{G}_1^*$ there is a $Q \in \mathbb{G}_2^*$ (and for all $Q \in \mathbb{G}_2^*$ there is a $P \in \mathbb{G}_1^*$, respectively) such that $e(P, Q) \neq 1$.
3. Computable, i.e., there is a probabilistic polynomial time algorithm that computes $e(P, Q)$ for all $(P, Q) \in \mathbb{G}_1 \times \mathbb{G}_2$.

Let P_1 be a generator of \mathbb{G}_1 and let P_2 be a generator of \mathbb{G}_2 , i.e., $\langle P_1 \rangle = \mathbb{G}_1$ and $\langle P_2 \rangle = \mathbb{G}_2$. The pairing e is called admissible if $e(P_1, P_2) = g$ such that $\langle g \rangle = \mathbb{G}_T$.

We denote with $\text{GenPair}(1^l) \rightarrow (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, P_1, P_2, e)$ an algorithm that on input a security parameter $l \in \mathbb{N}$ generates three groups \mathbb{G}_1 , \mathbb{G}_2 and \mathbb{G}_T of large prime exponent q , two generators $\langle P_1 \rangle = \mathbb{G}_1$ and $\langle P_2 \rangle = \mathbb{G}_2$ and an admissible pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$.

2.2.2 Intractability Assumptions

Decisional Diffie-Hellman Assumption

Definition 2.2 (DDH Assumption [42]). Let $l \in \mathbb{N}$ be a security parameter, $pk_e \leftarrow \text{GenPair}(1^l)$ where $pk_e = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, P_1, P_2, e)$ (Definition 2.1), $x, y \xleftarrow{\$} \mathbb{Z}_q$, $X \leftarrow xP_1$, $Y \leftarrow yP_1$ and $Z \xleftarrow{\$} \mathbb{G}_1$. The decisional Diffie-Hellman assumption in \mathbb{G}_1 is that every probabilistic polynomial time adversary \mathcal{A} has negligible advantage (in l)

$$\text{Adv}_{\mathcal{A}}^{\text{DDH}} = \left| \Pr [1 \leftarrow \mathcal{A}(pk_e, X, Y, xyP_1)] - \Pr [1 \leftarrow \mathcal{A}(pk_e, X, Y, Z)] \right|.$$

Bilinear LRSW Assumption

Definition 2.3 (Bilinear LRSW Assumption [42]). Let $l \in \mathbb{N}$ be a security parameter, $pk_e \leftarrow \text{GenPair}(1^l)$ where $pk_e = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, P_1, P_2, e)$ (Definition 2.1), $x, y \xleftarrow{\$} \mathbb{Z}_q$, $X \leftarrow xP_2$ and $Y \leftarrow yP_2$. Moreover, let $\mathcal{O}_{X,Y}$ be an oracle that on input $a \in \mathbb{Z}_q$ outputs a triple $(D, yD, (x + axy)D)$ where $D \xleftarrow{\$} \mathbb{G}_1$. Let Ω be the set of oracle queries made to $\mathcal{O}_{X,Y}$. The bilinear LRSW assumption is that for every probabilistic polynomial time

adversary \mathcal{A} and every $(a, D, E, F) \in [\mathcal{A}^{\mathcal{O}_{X,Y}}(pk_e, X, Y)]$ it holds that

$$\Pr [a \notin \Omega \wedge a \in \mathbb{Z}_q^* \wedge D \in \mathbb{G}_1 \wedge E = yD \wedge F = (x + axy)D]$$

is negligible in l .

2.3 Cryptographic Primitives

2.3.1 Pseudo-random Functions

Let $l \in \mathbb{N}$ be a security parameter, $\kappa, \alpha, \beta \in \mathbb{N}$ be polynomially bounded in l and let $F : \{0, 1\}^{\kappa+\alpha} \rightarrow \{0, 1\}^\beta$ be a family of functions. Consider the following security experiment $\mathbf{Exp}_{\mathcal{A}_{\text{prf}}}^{\text{prf-}b}$, where an adversary \mathcal{A}_{prf} interacts with a *PRF-challenger* \mathcal{C}^{prf} . When initialized with l, κ, α, β and $b \xleftarrow{\$} \{0, 1\}$, \mathcal{C}^{prf} chooses $k \xleftarrow{\$} \{0, 1\}^\kappa$ and initializes an oracle \mathcal{O}^{F_k} that on input $x \in \{0, 1\}^\alpha$ returns $y \leftarrow F_k(x)$ if $b = 1$ and $y \xleftarrow{\$} \{0, 1\}^\beta$ otherwise. After a polynomial number of queries to oracle \mathcal{O}^{F_k} , \mathcal{A}_{prf} must return a bit b' . \mathcal{A}_{prf} wins the security experiment if $b = b'$.

Definition 2.4 (Pseudo-random Function [72]). *A pseudo-random function (PRF) is a family of functions F with the following properties:*

1. *Each function $F_k \in F$ can be identified by a unique index $k \in \{0, 1\}^\kappa$.*
2. *There is a polynomial time algorithm that, given an index $k \in \{0, 1\}^\kappa$ and input $x \in \{0, 1\}^\alpha$, computes $F_k(x)$.*
3. *Each probabilistic polynomial time adversary \mathcal{A}_{prf} has at most negligible advantage*

$$\mathbf{Adv}_{\mathcal{A}_{\text{prf}}}^{\text{prf}} = |\Pr [\mathbf{Exp}_{\mathcal{A}_{\text{prf}}}^{\text{prf-}1} = 1] - \Pr [\mathbf{Exp}_{\mathcal{A}_{\text{prf}}}^{\text{prf-}0} = 1]|.$$

2.3.2 Random Oracles

Definition 2.5 (Random Oracle [17]). *Let $\alpha, \beta \in \mathbb{N}$. A random oracle RO is a probabilistic polynomial time algorithm RO that for each input $a \in \{0, 1\}^\alpha$ returns a uniformly random output $b \in \{0, 1\}^\beta$. More precisely, RO starts with an empty look-up table τ . When queried with an input a , RO first checks if it already knows a value $b = \tau(a)$. If this is not the case, RO chooses a uniformly random value $b \xleftarrow{\$} \{0, 1\}^\beta$ and sets $\tau(a) := b$. Finally, RO returns $\tau(a)$.*

2.3.3 Hash Functions

Definition 2.6 (Hash Function). *Let $\kappa, \alpha, \beta \in \mathbb{N}$. A family of hash functions $H := \{\text{Hash}_k : \{0, 1\}^\alpha \rightarrow \{0, 1\}^\beta \mid k \in \{0, 1\}^\kappa\}$ is a set of functions Hash_k indexed by k mapping bit strings of arbitrary but finite length α to bit strings of a fixed length β .*

Definition 2.7 (Collision-resistance [47]). *A hash function family H (Definition 2.6) is collision resistant if it is computationally infeasible to find any pair of inputs (a, a') with $a \neq a'$ such that $\text{Hash}_k(a) = \text{Hash}_k(a')$ for some $k \xleftarrow{\$} \{0, 1\}^\kappa$, i.e., for any probabilistic polynomial time adversary \mathcal{A} it holds that*

$$\Pr \left[\text{Hash}_k(a) = \text{Hash}_k(a') \mid (a, a') \leftarrow \mathcal{A}^{\text{Hash}} \wedge k \xleftarrow{\$} \{0, 1\}^\kappa \right] \leq \epsilon(\kappa, \alpha, \beta)$$

is negligible in κ, α and β .

2.3.4 Encryption Schemes

Definition 2.8 (Symmetric Encryption Scheme). *Let M be the message space and C be the ciphertext space. A symmetric-key encryption scheme is a tuple of algorithms $(\text{Genkey}, \text{Enc}, \text{Dec})$ where Genkey is the key generation, Enc is the encryption and Dec is the decryption algorithm where for all security parameters $l \in \mathbb{N}$ and every plaintext $m \in M$ it holds that*

$$\Pr \left[\text{Dec}(k; \text{Enc}(k; m)) = m \mid k \leftarrow \text{Genkey}(1^l) \right] = 1.$$

Definition 2.9 (Public-key Encryption Scheme). *Let M be the message space and C be the ciphertext space. A public-key encryption scheme is a tuple of algorithms $(\text{Genkey}, \text{Enc}, \text{Dec})$ where Genkey is the key generation, Enc is the encryption and Dec is the decryption algorithm where for all security parameters $l \in \mathbb{N}$ and every plaintext $m \in M$ it holds that*

$$\Pr \left[\text{Dec}(sk; \text{Enc}(pk; m)) = m \mid (sk, pk) \leftarrow \text{Genkey}(1^l) \right] = 1.$$

Definition 2.10 (Homomorphic Encryption Scheme [149, 74]). *Let M be the plaintext space and C be the ciphertext space of an encryption scheme. Moreover, let $\circ : M \rightarrow M$ and $\bullet : C \rightarrow C$ be two functions operating on elements of M and C , respectively. A public-key encryption scheme (Definition 2.9) is called homomorphic if for every pair of*

ciphertexts $c_1 = \text{Enc}(pk; m_1)$ and $c_2 = \text{Enc}(pk; m_2)$ with $c_1, c_2 \in C$ and $m_1, m_2 \in M$ it holds that $c_1 \bullet c_2 = \text{Enc}(pk; m_1 \circ m_2)$. The definition of a symmetric-key encryption scheme (Definition 2.8) with homomorphic properties is analogous. We indicate homomorphic encryption schemes by $(\text{Genkey}^h, \text{Enc}^h, \text{Dec}^h)$.

Real-or-random Indistinguishability

Definition 2.11 (Real-or-random Indistinguishability [15]). Let $(\text{Genkey}, \text{Enc}, \text{Dec})$ be an encryption scheme (Definition 2.9 or 2.8). Moreover, let $(sk, pk) \leftarrow \text{Genkey}(1^l)$ (resp. $k \leftarrow \text{Genkey}(1^l)$) for some security parameter $l \in \mathbb{N}$. Further, let $\mathcal{O}_{\text{RoR}}^b$ be an oracle that when queried with a plaintext $m \in M$ returns either $\text{Enc}(pk; m)$ (resp. $\text{Enc}(k; m)$) if $b = 0$ and $\text{Enc}(pk; m')$ (resp. $\text{Enc}(k; m')$) for a randomly chosen message $m' \in M$ if $b = 1$. An encryption scheme is said to be real-or-random indistinguishable if every probabilistic polynomial time adversary \mathcal{A} has at most negligible advantage (in l)

$$\text{Adv}_{\mathcal{A}}^{\text{RoR}} = \left| \Pr \left[1 \leftarrow \mathcal{A}^{\mathcal{O}_{\text{RoR}}^0} \right] - \Pr \left[1 \leftarrow \mathcal{A}^{\mathcal{O}_{\text{RoR}}^1} \right] \right|.$$

Chosen-plaintext Security

Definition 2.12 (CPA-Security [73, 16]). Let $(\text{Genkey}, \text{Enc}, \text{Dec})$ (Definition 2.8 or 2.9) be an encryption scheme with security parameter $l \in \mathbb{N}$. Consider the following security experiment $\text{Exp}_{\mathcal{A}}^{\text{CPA}-b}$: An algorithm \mathcal{C}^{cpa} (called CPA-challenger) generates an encryption key $k \leftarrow \text{Genkey}(1^l)$ (resp. a key pair $(sk, pk) \leftarrow \text{Genkey}(1^l)$ where pk is given to the adversary \mathcal{A}). The adversary \mathcal{A} must respond with two messages $m_0 \in M$ and $m_1 \in M$. \mathcal{C}^{cpa} then encrypts $c_b \leftarrow \text{Enc}_k(m_b)$ (resp. $c_b \leftarrow \text{Enc}_{pk}(m_b)$) and returns the resulting ciphertext c_b to \mathcal{A} , who now must return a bit b' that indicates whether c_b encrypts m_0 or m_1 . The result of the security experiment is b' , i.e., $\text{Exp}_{\mathcal{A}}^{\text{CPA}-b} = b'$. The encryption scheme is CPA-secure if every probabilistic polynomial time adversary \mathcal{A} (which we denote as CPA-distinguisher) has at most negligible advantage (in l)

$$\text{Adv}_{\mathcal{A}}^{\text{CPA}} = \left| \Pr \left[\text{Exp}_{\mathcal{A}}^{\text{CPA}-0} = 1 \right] - \Pr \left[\text{Exp}_{\mathcal{A}}^{\text{CPA}-1} = 1 \right] \right|.$$

3 Background on RFID

Radio frequency identification (RFID) enables RFID *readers* to perform fully automatic wireless identification of objects that are labeled with RFID *tags*. Initially, this technology was mainly used for electronic labeling of pallets, cartons and products to enable seamless supervision of supply chains. Today, RFID technology is widely deployed to many other applications as well, including animal identification [9, 177], library management [135], access control [9, 177, 141], electronic tickets [141, 177], passports [93] and even human implantation [96].

As pointed out in previous publications [196, 96], this prevalence of RFID technology introduces various risks, in particular concerning the privacy of its users and holders. The most deterrent privacy risk concerns the tracking of users. Thus, an RFID system should provide *anonymity* (confidentiality of the tag identity) as well as *untraceability* (unlinkability of the communication of a tag) even in case the state of (i.e., the information stored on) the tag has been disclosed. RFID applications in practice must also achieve various security and functional goals. The security goals include *device* and/or *user authentication*, which prevents the adversary from impersonating and forging tags, and *availability*, which means the resilience to remote tampering that allows denial-of-service attacks. The functional goals include *efficiency*, including fast verification of cost-efficient tags and *scalability*, such as the support of a large number of tags.

Most currently used RFID systems do not offer privacy at all [9, 141, 177]. This is mainly because current cost-efficient tags do not provide the necessary computational resources to run privacy-preserving protocols, which heavily rely on public-key cryptography. Moreover, as we point out later in Section 3.3, privacy-preserving solutions without public-key cryptography do not fulfill important security and/or functional requirements and thus, are inapplicable to real-world applications.

Remark. Parts of this chapter have been published in [166] and [85].

3.1 System Architecture and Requirements

In this section, we describe the actors of an RFID system and specify the requirements that such a system should satisfy in order to be considered secure.

3.1.1 Common System Architecture

A typical RFID system consists of at least an *issuer* \mathcal{I} , a *tag* \mathcal{T} and a *reader* \mathcal{R} that is used to communicate to the tags (Figure 3.1). \mathcal{I} is the entity that enrolls and maintains the RFID system. Hence, \mathcal{I} initializes \mathcal{T} and \mathcal{R} before they are deployed in the system. \mathcal{T} and \mathcal{R} are called *legitimate* if they have been initialized by \mathcal{I} .

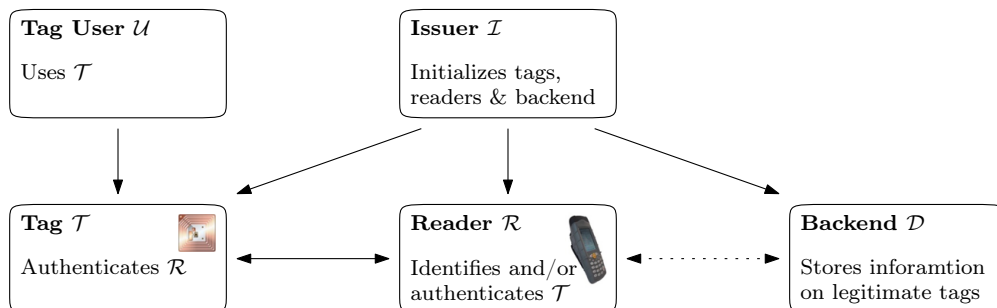


Figure 3.1: Typical RFID System Architecture

A tag \mathcal{T} is an integrated circuit embedded into a plastic card or sticker that is attached to the object to be identified [75, 141], e.g., a user who uses the tag to prove that he is eligible to access some service. In many applications \mathcal{T} is a hardware equipped with a radio interface. Currently available passive¹ RFID devices are powered by readers and thus cannot initialize communication, have limited memory, are not tamper-resistant and limited to basic cryptographic computations, including keyed hashing, symmetric-key encryption and random number generation [9, 141]. All information, e.g., the secrets and data stored on \mathcal{T} is denoted as the *state* of \mathcal{T} .

The reader \mathcal{R} is a stationary or mobile computing device that interacts with \mathcal{T} when \mathcal{T} gets into the reading range of \mathcal{R} . The main purpose of this interaction usually is the authentication of \mathcal{T} to \mathcal{R} . Depending on the use case, \mathcal{R} may also authenticate to \mathcal{T} and/or obtain additional information such as the identity of \mathcal{T} . \mathcal{R} can have a

¹Active RFID devices have an on-tag power supply and thus are too expensive for most commercial applications.

sporadic or permanent online connection to some backend system \mathcal{D} , which typically is a database maintaining detailed information on all tags in the system. \mathcal{D} is initialized and maintained by \mathcal{I} and can be read and updated by \mathcal{R} .

Today, RFID is mainly used for *identification* and *authentication* purposes including access control [96] and anti-counterfeiting systems [184]. Informally, identification means the process of recognizing which entity one is communicating to while authentication means the (cryptographic) corroboration of the identity of this entity.

Users of an RFID system own one or more tags that can be interrogated without optical or physical contact. This greatly enhances convenience in access control systems since users do not need to insert their security token into a reader but can leave it in their wallets or pockets. However, wireless interaction is imperceptible and thus may allow unauthorized entities to obtain user-related data including personal information and locations. As a consequence, in addition to the threats to conventional authentication systems, RFID systems must consider privacy and security problems that are related to the wireless radio interface.

3.1.2 Attacks on RFID Systems

The main goal of every authentication scheme, including RFID, is to prevent unauthorized users from cheating an honest reader in order to obtain unauthorized access. Beyond guaranteeing this main goal, there are other, subtle attacks against RFID systems that do not only aim at creating user profiles but are crucial for the deployment of RFID systems to real-world applications.

Impersonation

The most obvious attack against RFID systems is motivated by entities, who want to gain unauthorized access. Specifically, the adversary must obtain or simulate a tag that is accepted by an honest reader. To achieve this, the adversary may perform various attacks, including man-in-the-middle and replay attacks, against the underlying authentication protocols, and he may attempt to create forged tags and copy legitimate tags of honest users.

Tracing

A more subtle attack aims at obtaining information on users and their movements. When using conventional authentication protocols, a tag can be easily identified and traced during verification. Moreover, if users can be identified when obtaining a tag (e.g., when using an identifying payment method such as a credit card for buying an RFID-enabled e-ticket), the issuer of the tag can link the corresponding tag to the identity of its owner. Since the issuer and the readers are typically under the control of the same entity (e.g., the transit company in the case of electronic transit tickets), this results in a complete loss of the user's location privacy. For instance, the operator of the readers may link the transactions of the user's tags and correlate this data with the geographical location of the readers.

Denial-of-Service

Another type of adversary may want to harm (e.g., to blackmail) the company running the RFID system by disturbing the authentication process of honest users. Besides financial losses such an attack would seriously damage the reputation of the affected company and thus should be prevented by every dependable RFID system. However, since tags are wireless devices that can be attacked unnoticeably, the adversary may try to exploit deficiencies of the protocols such that a tag is no longer accepted by honest readers.

Depending on the underlying use case and business model, RFID protocols must be carefully designed to prevent some or all of these attacks. Section 3.1.3, introduces different trust and adversary models for RFID systems. A complete list of requirements for practical privacy-preserving RFID systems is given in Section 3.1.4.

3.1.3 Trust and Adversary Models

In an ideal setting, no entity must be trusted. However, in practice, at least the issuer must be trusted to only create tags for eligible users. Moreover, each reader must be trusted to accept only those tags that have been issued by a genuine issuer. These are reasonable assumptions since in practice the issuing entity and the readers are typically physically controlled by the same entity (e.g., the transit company in the case of e-tickets) or share the same goals. However, this entity may be curious and collect user information.

Ideally, users should be anonymous to every entity, including the tag issuer and all readers. However, due to technical constraints this is not always feasible in practice.² Thus, a reasonable trust model for a practical solution is that users must at least trust the issuer and, depending on the implementation, also all readers. Obviously, a trust model which only requires the tag issuer to be trusted is preferable.³

Summing up, while tag issuers and readers must trust each other, for users there are three possible trust models:

- TM1: Users do not need to trust any entity.
- TM2: Users must trust *only* the tag issuer.
- TM3 Users must trust the tag issuer *and* all readers.

To realize trust model TM1, the RFID scheme must provide full anonymity. However, this seems to be related to other systems such as anonymous credentials and thus, TM1 seems to be possible only with high computational and communication resources, which is inappropriate for low-cost RFID devices.

In trust model TM2 users must at least trust the issuer with respect to their privacy whereas in trust model TM3 users must additionally trust the readers. Privacy to all entities outside the system (i.e., all unknown entities that are not trusted by the user) must be preserved in any case. Trust models TM2 and TM3 can be achieved by existing RFID protocols. However, as discussed later in Section 3.3, these protocols lack usability and thus are not applicable to most real-world scenarios.

A common assumption in RFID systems is that all communication that takes place during the process of issuing a tag cannot be eavesdropped or manipulated by the adversary. This is reasonable in practice since a user may either use out-of-band communication or a secure channel to communicate to the tag issuer. However, following the traditional adversarial models, the adversary can eavesdrop all communication of a tag after it has been issued, even from outside the nominal reading range of the tags [111]. Moreover, the adversary may perform active attacks on the corresponding protocols, which means

²Note that very powerful attackers could trace tags and their users, e.g., by observing them using a large number of cameras. However, since such an attack is also possible against persons without tags, an RFID system cannot protect against such powerful attacks.

³Note that in this case, user information is managed only by one single known entity that can be committed by law to the confidential use of the collected user data and that can be monitored by means of inspections (similar observations hold for credit card companies).

that he can interact with all parties at the protocol level. Furthermore, most commercial RFID tags are cost-efficient devices without expensive protection mechanisms against physical tampering [9, 141]. Hence, the adversary can physically attack (*corrupt*) a tag and obtain its state, e.g., its secrets. Besides the communication between tag and reader, the adversary can also obtain useful auxiliary information (e.g., by visual observation) on whether the reader accepted the tag as a legitimate tag [99, 190].

In practice, RFID readers are embedded systems that can be integrated into mobile devices (e.g., mobile phones or PDAs) and computers. The resulting complexity exposes readers to sophisticated hard- and software attacks, e.g., viruses and Trojans. This problem aggravates for mobile readers that can easily be lost or stolen. Hence, the adversary can get full control over the reader [12, 65, 139], which is captured in trust models TM1 and TM2.

In trust model TM1, the issuer may be compromised by an adversary who wants to violate privacy. In all other trust models the adversary cannot corrupt the issuer.

3.1.4 Requirement Analysis

We formally describe the requirements of a dependable RFID system, where crucial security, privacy and usability properties have to be simultaneously achieved.

Security Goals

As mentioned in Section 3.1.2, one of the most important security goals is *tag authentication*. Thus no unauthorized user (i.e., who is not in possession of a legitimate tag) should be able to cheat any honest reader in order to obtain an unauthorized access. This includes creating illegitimate (*forged*) tags that are accepted by honest readers and emulating (*impersonation*) or copying (*cloning*) legitimate tags. Most use cases (such as access control systems) additionally require the reader to determine the authentic tag identity (*tag identification*). Another major requirement for any tag-based authentication scheme is the resilience to remote tampering with tags that permanently prevent users from using the RFID system (*denial-of-service*) [35]. Moreover, there are several applications (e.g., electronic tickets) where reader authentication is a fundamental security property. However, there are also use cases (e.g., electronic product labels) that do not require reader authentication. We summarize the security goals with respect to the different adversary models described in Section 3.1.3 as follows:

- *Tag authentication*: The adversary should not make an honest reader accept.
- *Reader authentication*: The adversary should not make an honest tag accept.
- *Unclonability*: It is infeasible for the adversary to duplicate a legitimate tag.
- *Availability*: It is infeasible for the adversary to alter tags by interacting with them over the wireless interface.

Privacy Goals

Since RFID enables efficient detection and identification of a large number of tags, detailed user profiles (including personal data and movements) can be created. In particular, the wireless interface of RFID tags allows the adversary to communicate to the tags without consent or knowledge of the tag user. The problem aggravates if tags can be associated with the identity of their corresponding users since this results in a complete loss of location privacy. Thus, to ensure location privacy, an RFID system must fulfill the following requirements with respect to the different adversary models described in Section 3.1.3:

- *Confidentiality*: It is infeasible for the adversary to access user data.
- *Anonymity*: It is infeasible for an adversary to identify tags, even when the state of the tag has been disclosed.
- *Unlinkability*: It is infeasible for the adversary to trace tags, even when the state of the tag has been disclosed.

Note that inexpensive RFID tags usually cannot provide expensive tamper-resistant hardware and thus, the adversary in practice can obtain the internal state of (i.e., all information stored on) a tag. Therefore, a stronger notion of location privacy is needed to capture traceability of tags in this case. To distinguish traceability in past or future protocol runs, [118] considers the notion of *forward* and *backward traceability*:

- *Backward traceability*: It should be infeasible for an adversary with access to the current state of a tag to trace this tag in *previously recorded* protocol runs.
- *Forward traceability*: It should be infeasible for an adversary with access to the current state of a tag to trace this tag in *future* protocol runs.

Functional Requirements

In addition to the privacy and security goals it is important to consider some functional requirements desired for many real-world applications.

First, the manufacturing costs of a tag should be minimal, which means that the computational and storage requirements on the tags should be as low as possible. Additionally, verification of tags must be fast. For instance, it should be possible to verify a tag while a user is walking by or shortly holding his tag near a reader (e.g., verification of an RFID e-ticket should be possible while entering a bus). Therefore, corresponding RFID protocols must be designed carefully to minimize the amount of computation and communication that must be performed without lowering the security and privacy requirements discussed above. Moreover, an RFID system should be able to handle a large number of tags. We summarize these goals as follows:

- *Efficiency*: Verification of tags must be fast.
- *Scalability*: A large number of tags must be supported.

Depending on the underlying application and the technological constraints, a practical realization may not be able to fulfill all of these goals and requirements. In particular, the security and functional requirements often contradict the privacy requirements.

Further, it seems that in many RFID applications, the requirements are prioritized. First, the deployed solution should be *correct*, i.e., it should be ensured that every legitimate tag is accepted by every honest reader. Second, the system should be *available*, i.e., achieve a certain robustness against denial-of-service attacks. Third, the system should be secure and finally, protect the privacy of its users.

3.2 Existing Security and Privacy Models

It is essential to carefully formalize the security and privacy goals discussed in Section 3.1.4 to enable the design of secure and privacy-preserving RFID protocols that are also usable in practice. The existing literature proposes various security and privacy models for RFID. One of the first privacy definitions for RFID has been proposed by Ohkubo et al. [143] and captures leakage of information on user-related data, including identities movements of users. This definition is based on a security experiment where

the adversary is challenged to distinguish a random value from the output of a legitimate tag. Ohkubo et al.'s definition [143] also covers backwards traceability. However, this privacy model does not consider adversaries who can modify tags (e.g., by manipulating the tag memory) in order to trace them.

Juels [95] introduces a very restrictive adversary model specifically for RFID tags that cannot perform cryptographic operations. This model is based on assumptions on the number of queries the adversary can make to a tag and aims at defining privacy to a broad range of real-world attacks. However, it does not allow the adversary to corrupt tags and thus does not capture forward and backwards untraceability. Ouafi et al. [147, 147] present an RFID privacy model that defines privacy based on a security experiment similar to the privacy definition of Juels and Weis [99, 100].

Avoine et al. [10, 11] propose a security and privacy model that provides various flexible definitions for different levels of privacy based on a security experiment where the adversary must distinguish two known tags. Juels and Weis [99, 100] extend this model by introducing the notion of *side-channel information* that reveals whether authentication of a tag was successful or not. However, their extended model does not capture backwards traceability since it does not allow the adversary to corrupt tags. Damgård and Pedersen [45, 46] extend Juels' and Weis' definition [99, 100] by adding a *completeness* and *soundness* requirement, which means that a reader must accept *all* but *only* legitimate tags. This definition has been further improved by Ha et al. [78] and refined [189, 123, 115] to consider backwards traceability.

Another approach to define privacy based on the universal composability (UC) framework [41] has been presented by Burmester et al. [34, 35]. This model claims to be the first that considers availability, which means that it captures security against denial-of-service attacks. However, it does not consider the privacy of corrupted tags, i.e., backwards and forward traceability.

Vaudenay [190] presents a privacy definition that generalizes and classifies previous RFID privacy models by defining eight levels of privacy that correspond to different adversary models. The strongest adversary model covers all notions of privacy of previous works covering side-channel information, privacy of corrupted tags and adversaries that can interact with tags and manipulate them at the protocol level. Moreover, the security definition is equivalent to the definition by Damgård and Pedersen [45, 46]. Paise and Vaudenay [150] later extended Vaudenay's original model [190] to additionally consider

authentication of readers to tags whereas Ng et al. [137] showed that the eight privacy classes defined by Paise and Vaudenay [190, 150] can be reduced to three privacy classes under some restrictions on the power of the adversary. Canard et al. [40] present an RFID security and privacy model based on Vaudenay’s model [190] where privacy of tags is defined based on the adversary’s capability to distinguish a corrupted tag in different protocol runs, which is comparable to forward and backwards untraceability.

Another privacy model [23, 84] defines privacy based on a security experiment where the adversary must distinguish whether two transactions originate from the same or two different tags.

3.3 Analysis of Existing Solutions

This section surveys existing approaches to privacy in RFID systems in the literature. Specifically, we focus on the main weaknesses of existing solutions with respect to the security, privacy and functional requirements discussed in Section 3.1.4.

3.3.1 Physical Methods

There is a body of literature that proposes physical solutions to enhance the privacy of RFID systems. For instance, some RFID tags support a *kill command*, which is a tag-specific password programmed at manufacturing time that can be used to permanently disable the tag [75] such that it cannot be read any longer. This approach has been designed for electronic product labels that can be disabled after the corresponding product has left the supply chain and is given to the end-user. Another simple approach is to jam the radio interface of tags. The first solution is to put the tags into a Faraday cage, which is a container of metal mesh or foil that is opaque to radio signals (of a certain frequency). Today, RFID blocking wallets that integrate Faraday cages are widely available, e.g., to protect RFID-enabled passports from unauthorized reading. Alternatively, users may carry a special active jamming device that disturbs the radio signals of tags and readers in the user’s vicinity [153].

Since all of these more or less radical approaches permanently disable the tags or require the user to interact with them, they eliminate one of the main advantages of RFID. Thus we focus on more sophisticated solutions that enhance user privacy by protocol-based techniques without affecting the usability of RFID systems.

3.3.2 Anonymous Authentication Protocols

In an ideal RFID system, readers should learn nothing from the verification except that a tag is legitimate. So far, fully anonymous authentication of RFID tags to readers has been discussed only in a few papers [86, 21, 20]. The schemes proposed by Heydt-Benjamin et al. [86] and Bichsel et al. [20] employ anonymous credential systems [38, 39]. Heydt-Benjamin et al. [86] describe a generic anonymous payment system (which includes anonymous authentication) for RFID-powered public transport tickets based on anonymous credentials but they do not provide any implementation details. Bichsel et al. [20] present the implementation of a full fledged anonymous credential system on Java Cards, which are expensive contactless smartcards.⁴ Since the use of anonymous credentials implies high computational requirements (public-key cryptography) to all devices involved, these systems do not comply to the capabilities of most RFID systems in practice that require fast authentication of cost-effective tags (cf. Section 3.1.4). Thus, these techniques are not applicable unless powerful mobile computing devices (such as mobile phones or PDAs) are used.⁵ However, the use of mobile computing devices has its own risks. These devices may run out of power (which violates availability) and can be compromised by Trojans, which brings up new security challenges. Moreover, many users do not yet own an NFC-enabled mobile phone. An alternative approach to anonymous RFID-based payment has been proposed by Blass et al. [21]. However, this approach is inflexible since the total number of tags in the system must be fixed during system initialization. Further, it is not scalable since the amount of data that needs to be stored on the reader depends on the total number of tags in the system. Summing up, existing approaches to anonymous authentication of RFID tags are not applicable to most real-world RFID applications.

3.3.3 Privacy-preserving Authentication Protocols

There is a large body of literature on different approaches to implement privacy-preserving mechanisms for low-cost RFID tags. For instance, Juels [96] gives a comprehensive overview of different solutions. He classifies RFID tags as *basic tags* and *symmetric-key tags*. Basic tags refers to tags that have no computational and no cryptographic capa-

⁴Many RFID applications deployed in practice require RFID tags to be as cost-efficient as possible which renders contactless smartcards that typically cost more than 1–2 € per item as too expensive.

⁵An increasing number of mobile phones and PDAs supports the Near Field Communication (NFC) standard [136], which allows them to communicate to RFID tags.

bilities. Symmetric-key tags means tags that are capable of performing at least some symmetric cryptographic functions, such as random number generation, hashing and/or encryption. Moreover, there are tags that can perform basic computations but that are too constrained to execute cryptographic schemes, which are not considered in Juels' classification.

Protocols for Basic Tags

As basic tags cannot perform any cryptographic operation they cannot be used in applications that require strong authentication. These tags provide only wireless readable memory, i.e., they can only forward the data stored in their memory and are subject to replay and cloning attacks. This means that all the data stored on such a tag can be read and be used to create identical copies or to emulate the original tag to an honest reader. Another problem related to cloning is *swapping*. This means that the adversary can copy the data stored on tag *A* to another tag *B* and vice versa and thus change the identities of these tags. Therefore, basic tags cannot provide authentication and unclonability.

Moreover, many solutions to enhance the privacy of basic tags require the tags to provide *many-writable* memory [97, 74, 7]. The basic idea of these schemes is to frequently update the information stored on the tags such that the adversary cannot trace them. However, due to the lack of secure access control mechanisms it is impossible to prevent unauthorized writes to such tags. A simple denial-of-service attack is to write random data to a tag, which makes an honest reader no longer accepting the tag until it is re-initialized with correct data. This clearly violates the availability requirement. Moreover, the adversary could *mark* tags (e.g., store some recognizable data on them) such that he can track the tags even if they are frequently updated [7]. Obviously, this violates location privacy.

As a consequence, tags without any cryptographic functionality cannot be used in applications that require authentication. Thus, it is inevitable to use tags that are capable of performing at least some cryptographic functions if authentication is of concern.

Protocols for Computational Tags

Computational tags can perform basic arithmetic operations but cannot execute complex cryptographic schemes. The most prominent class of privacy-preserving protocols for this kind of tags are the HB protocols. The original HB protocol [90] is secure against

passive adversaries [98]. HB^+ [98] is a variant of HB that is claimed to be secure against active adversaries, even when multiple parallel and concurrent protocol-runs are performed [105, 106]. However, HB^+ is vulnerable to man-in-the-middle attacks [69, 117] that allow extracting the authentication secrets of tags. $\text{HB}^\#$ [70] is a variant of HB^+ that is secure against man-in-the-middle attacks and has a lower communication complexity. Another variant of HB, HB^{++} [30], is secure against man-in-the-middle attacks but requires the tags to compute universal hash functions, which exceeds the capabilities of computational tags. This is addressed by Trusted-HB [29], a variant of HB^+ that uses lightweight hashing based on LFSRs [112]. However, Trusted-HB fails to achieve its claimed security goals [63]. While all these protocols are not secure against adversaries that can corrupt tags, PUF-HB [80] is a variant of HB that uses Physically Unclonable Functions (PUFs) to be secure against tag corruption.

A drawback of all HB protocol variants is their high communication complexity since they require many rounds of interaction between the tag and the reader. Moreover, they require the reader to perform an exhaustive search for the authentication secret of the authenticating tag. Further, the security evaluation of the HB protocols typically does not consider tag corruption.

Protocols for Symmetric-key Tags

A general problem of implementing privacy-preserving authentication based on symmetric-key cryptography is how to inform the other party which key must be used. Apparently, a tag cannot disclose its identity before the reader has been authenticated since this would violate unlinkability. Therefore, the reader does not know which authentication key it should use and thus cannot authenticate to the tag. Essentially there are two approaches that address this problem. The first approach allows the reader to efficiently find the key used by the tag whereas the second approach frequently updates the identity of tags in a way that allows the reader to efficiently deduce the initial tag identity. We now discuss both approaches more detailed.

Key search approach. The basic idea of this approach has been introduced by Weis et al. [196]: Let $F_k(m)$ be a pseudo-random function (Definition 2.4) on some message m using some key k . To authenticate to a reader, the tag first computes $h \leftarrow F_k(r)$, where k is a tag-specific secret key and r is a random value chosen by the tag. On receipt of

(h, r) , the reader forwards this tuple to a trusted server that computes $h_i \leftarrow F_{k_i}(r)$ for all keys $k_i \in K$, where K denotes the set of the keys of all legitimate tags. The server accepts the tag if it finds a $k_i \in K$ such that $h_i = h$. Finally, the server sends its decision whether to accept or to reject the tag to the reader. Since r is randomly chosen each time the tag is queried, the tag always emits a different tuple (h, r) which cannot be linked to the tuples sent in previous protocol runs. Moreover, the reader does not learn the identity (i.e., the key k) of the tag since it only receives the response from the server. An obvious drawback of this solution is that the computational cost for the server to verify a tag is linear in the number of legitimate tags. Therefore, this basic approach does not fulfill the efficiency and scalability requirements (cf. Section 3.1.4). Another disadvantage of this approach is that all readers must have an online connection to the server, which, depending on the use case, may not be practical. Moreover, the tag must trust the server with respect to its privacy since the server can identify the tag when it finds the right key. Furthermore, this solution provides no security against replay-attacks since the adversary may impersonate the tag by replaying any previously recorded tuple (h, r) , which violates authentication.

There are many subsequent works [135, 55, 118, 176] that follow and optimize this approach by introducing new setup assumptions or by lowering the security and/or privacy requirements. For instance, Molnar and Wagner [135] improve the key search approach described above. The idea is to arrange the keys of all tags in a hierarchical tree. Each leaf of this tree corresponds to a tag, which means that all keys on the path from the root to the leaf are assigned to the corresponding tag. To authenticate to a reader, a tag runs one authentication protocol for each key it stores. Since all keys are arranged in a hierarchical tree, the reader does not need to search the whole key space. It is sufficient to search all keys of the first level of the subtree whose root is the key that has been used in the previous authentication protocol. Assume that the tree storing all keys has depth d and branching degree b . Then this protocol can handle at most $n = b^d$ tags and each tag must store d keys. Moreover, the verification of a tag requires the tag to run d authentication protocols with the reader. Compared to the basic approach [196], the reader has to perform only $b \cdot d$ instead of $n = b^d$ computations to verify one single tag. However, since this scheme requires the tags to share several keys, compromise of one tag violates the location privacy of others [11].

Tsudik [183] improves the key search approach by various pre-computations during

the creation of a tag. A tag is initialized with a tag-specific key k , a counter $t_i \leftarrow t_0$ and a maximum value t_{\max} for that counter. Moreover, $(t_0, F_k(t_0)), \dots, (t', F_k(t_{\max}))$, where F is a pseudo-random function (Definition 2.4), is stored in the database of the reader. To authenticate a tag, the reader sends t_j to the tag. In case t_j has already been used (i.e., $t_j < t_i$) or exceeds t_{\max} (i.e., $t_j > t_{\max}$), the tag returns a random value h_j . Otherwise, the tag responds with $h_j \leftarrow F_k(t_j)$ and updates $t_i \leftarrow t_j$. The reader accepts if it finds a tuple (t_j, h_j) in its database. According to Tsudik [183], this protocol does not provide security against denial-of-service attacks, which violates availability. Moreover, the protocol does not provide unlinkability [146] since it is possible to trace tags that have different maximum counter values. Clearly, this violates location privacy. Further, it does not provide authentication since the adversary may query a tag with several different t_j and learn the corresponding responses h_j which can later be replayed to an honest reader [146].

Identity-update approach. This approach relies on updating the identity of a tag each time it has been authenticated. Some of the protocols following this approach allow to authenticate a tag in constant time. However, these solutions require the readers to have permanent access to a trusted database that keeps track of the identity updates of all legitimate tags. As discussed above, this is inappropriate for many practical systems.

One of the first protocols following this approach is by Ohkubo et al. [144]. They propose a tag to update its state each time it is interrogated. Therefore, a tag is initialized with some initial identity ID_0 . The reader has access to a database that stores (k, ID_0) of each tag in the system. Each time a reader communicates to the tag, the tag responds with $r_i \leftarrow F_k(ID_i)$, where F is some one-way pseudo-random function (Definition 2.4). At the same time, the tag updates its identity to $ID_{i+1} \leftarrow F'_k(ID_i)$ where F' is a one-way pseudo-random function that is different from F . To identify the tag, the reader computes $r \leftarrow F_k(F'_k(ID_i))$ for each (k, ID_0) in the database and all $i \in \{0, \dots, m\}$ for some m . The reader accepts that tag only if it finds $r = r_i$. This means that, as in the basic key search approach [196], the verification of one single tag depends on the number of all legitimate tags in the system. Moreover, the maximum number of interrogations per tag is fixed to m . Thus, this protocol obviously does not fulfill the efficiency and scalability requirements (cf. Section 3.1.4). Moreover, the adversary may perform denial-of-service attacks since a tag can be invalidated by interrogating it more than m times, which clearly violates the availability requirement. However, this approach provides backwards

traceability. Since F and F' are one-way pseudo-random functions, an adversary who corrupted a tag cannot compute its preceding identities ID_i nor can he recognize the previous responses r_i of the tag.

Henrici and Müller [83] and Dimitriou [55] consider the problem of denial-of-service attacks and allow a tag to update its state only after the reader has been successfully authenticated to the tag. However, this allows tracing of tags between two successful authentications to a legitimate reader. The protocol proposed by Henrici and Müller [83] makes a tag to additionally transmit the number n of interactions since the last successful authentication to a legitimate reader. This information is used by the reader to speed up the identification of the tag and to prevent replay attacks. However, the adversary can trace tags by increasing the value n to an exceptionally high value that he can recognize later [10]. Thus, this approach does not provide location privacy.

Another protocol [134] assigns to each tag an authentication key k and identifier ID . The reader has access to a database that contains a tuple (ID, k) for each legitimate tag. To authenticate to a reader, the tag generates a random number r and sends its randomized identifier $e \leftarrow r \cdot k + ID$ to the reader. The reader accepts the tag if it finds a tuple (ID, k) in his database for which $ID = e \bmod k$. In the worst case, the reader must do this test for all tuples (ID, k) of each tag it knows. Clearly, this violates the efficiency and scalability requirements (cf. Section 3.1.4).

Anonymizer-enabled RFID Protocols

A promising approach to enhance the privacy in RFID systems without lifting the computational requirements on tags are anonymizer-enabled protocols, where external devices (*anonymizers*) are in charge of providing anonymity of tags. Anonymizer-enabled RFID protocols are very suitable for many practical scenarios with privacy needs that use cost-efficient tags. The main concept of existing anonymizer-enabled protocols [97, 74, 169, 7] is that each tag stores a ciphertext that encrypts the information carried by the tag (such as the tag identifier) under the public key of the reader. This ciphertext is sent to the reader each time the tag authenticates. Since this ciphertext is static data and can be used to track and to identify the tag, it must be frequently changed to provide anonymity and unlinkability. However, current RFID tags [9, 8, 141] are not capable of updating this ciphertext on their own and thus, privacy in these protocols relies on anonymizers that frequently refresh the ciphertexts stored on the tags. The first proposal

to use anonymizers [97] considers a plan by the European Central Bank to embed RFID tags into Euro banknotes to aggravate forgeries [181]. It proposes to store a ciphertext of the serial number of the banknote on an RFID tag attached to the banknote. Each time the banknote is spent, anonymizers in shops or banks re-encrypt the ciphertext stored on the tag. The drawback of this scheme is that the serial number of the banknote must be optically scanned before its ciphertext can be re-encrypted. Golle et al. [74] introduce a primitive called *universal re-encryption*, which is an extension of the El Gamal encryption scheme where re-encryption is possible without knowledge of the corresponding (private and public) keys. However, in this scheme the adversary can *mark* tags such that he can recognize them even after they have been re-encrypted. This issue has been addressed by Saito et al. [169] who show tracing attacks and propose solutions. Ateniese et al. [7] improve the ideas by Golle et al. [74] and Saito et al. [169] by introducing the notion of *insubvertible encryption*, which adds a signature on the blinded public key of the reader that is linked to the static ciphertext stored on the tag. Re-randomization involves this signature in a way that prevents the adversary from marking tags. However, the data stored on the tag can be easily replayed and copied to another tag.

All known anonymizer-enabled schemes are subject to impersonation attacks since authentication is only based on the ciphertext that the tag sends to the reader. Moreover, existing security models do not capture RFID systems that use anonymizers.

3.4 Conclusion and Open Problems

Existing solutions for privacy-preserving authentication in RFID systems either do not provide privacy in the presence of real-world adversaries who can corrupt tags or they suffer from drawbacks such as the possibility of denial-of-service and impersonation attacks as well as inefficient tag verification, which prevents their deployment in practice. Hence, the design of provably-secure and privacy-preserving authentication protocols for RFID that are applicable to real-world scenarios is a challenging open problem.

4 RFID Security and Privacy Revisited

In this chapter, we analyze one of the most comprehensive RFID security and privacy models [190, 150], which generalizes and improves many previous works. We point out weaknesses and deficiencies in this model and investigate some subtle issues such as tag corruption aspects that lead to the impossibility of achieving both mutual authentication and any reasonable notion of privacy in their model. Specifically, we show that in this model it is *impossible* to achieve *any* notion of privacy simultaneously with reader authentication (under full state disclosure) except for the weak and narrow-weak privacy notions. As a consequence, two of the protocols given in [150] do not achieve their claimed privacy properties. Our results led to the refinement of the Paise and Vaudenay RFID security and privacy model [191] and they were considered in subsequent works on privacy-preserving RFID systems [84, 53].

Remark. The results presented in this chapter are due to the author of this work and the result of many intensive discussions with Frederik Armknecht (University of Mannheim, Germany), Ahmad-Reza Sadeghi (TU Darmstadt, Germany), Alessandra Scafuro (University of Los Angeles, USA) and Ivan Visconti (University of Salerno, Italy). Parts of this chapter have been published in [6] and in [5].

4.1 Motivation and Contribution

The design of a secure privacy-preserving RFID scheme requires a careful analysis in an appropriate formal model. There is a large body of literature on security and privacy models for RFID [10, 99, 35, 190, 150, 51] (cf. Section 3.2). Existing solutions often do not consider important aspects such as adversaries with access to auxiliary information, e.g., on whether the identification of a tag was successful or the privacy of corrupted tags whose state has been disclosed. In particular, tag corruption is usually considered to happen only *before* and/or *after* but *not during* a protocol-run. However, in practice there are a variety of side-channel attacks [130, 91, 104] that extract the state of a tag based on the observation of, e.g., the power consumption of the tag *while* it is executing a protocol with

the reader. Since RFID tags are usually cost-effective devices without expensive tamper-proof mechanisms [9, 141], tag corruption is an important aspect to be covered by the underlying (formal) security model. Though in the literature, tag corruption during protocol execution is rarely considered. To the best of our knowledge the only security and privacy model that considers corruption of tags during protocol executions is the one by Burmester et al. [35]. However, this model does not consider issues like the privacy of tags after they have been corrupted and privacy against adversaries with access to auxiliary information. Tag corruption during protocol-runs has been informally discussed by Damgård and Pedersen [51]. However, their formal RFID security and privacy model assumes that such attacks cannot occur. Moreover, they indicate informally without giving formal proofs that tag corruption during protocol execution may have an impact on the formal definitions of the security and privacy model by Paise and Vaudenay [190, 150], which is the basis for many subsequent works [137, 138, 32, 165, 164, 50, 49, 40, 168, 167] (cf. Section 3.2). The first papers [50, 49] addressing tag corruption during protocol-runs in Vaudenay’s model [190] show that privacy can be achieved under the assumption that tag corruption during the protocol execution can be detected by the tag.

In this chapter, we focus on the security and privacy model by Paise and Vaudenay [150] (that is based on Vaudenay’s model [190]), which we call the *PV-Model* (Paise-Vaudenay Model) in the following. The PV-Model is one of the most comprehensive RFID security and privacy models up to date since it captures many aspects of real world RFID systems and aims at abstracting most previous works in a single concise framework. It defines mutual authentication between RFID tags and readers and several privacy notions that correspond to adversaries with different tag corruption abilities. However, as we show in this chapter, the PV-Model suffers from subtle deficiencies and weaknesses that are mainly caused by tag corruption aspects. In the PV-Model, each tag maintains a state that can be divided into a persistent and a temporary part.¹ The *persistent state* subsumes all information that must be available to the tag in more than one interaction with the reader (e.g., the authentication secret of the tag) and that can be updated during the interaction with the reader. The *temporary state* consists of all ephemeral information that is discarded by the tag after each interaction with the reader (e.g., the randomness used by the tag). In the PV-Model it is impossible to achieve any notion of privacy that allows tag corruption if the adversary can obtain *both* the persistent *and* the temporary

¹During a protocol execution tags could store some temporary information that allows them to verify the response of the reader.

tag state by tag corruption [150]. This issue is addressed in the PV-Model by the assumption that each tag erases its temporary state each time it gets out of the reading range of the adversary. However, this assumption leaves open the possibility to corrupt a tag *while* it is in the reading range of the adversary, i.e., *before* its temporary state is erased. In particular, the PV-Model allows the adversary to corrupt a tag *while* it is executing the authentication protocol with the reader.

Furthermore, an adversary in practice could physically tamper with a tag such that the tag resets its state and randomness to a previous value, which is not considered in the privacy notions of the PV-Model. It has been shown [5] that, by extending the PV-Model to capture reset attacks on tag states and randomness, no privacy can be achieved and that, when tags are stateless (i.e., when tags cannot update their persistent state), then destructive privacy is impossible.

In this chapter, we point out subtle weaknesses and deficiencies in the PV-Model. First, we show that the assumption of erasing temporary tag states whenever a tag gets out of the reading range of the adversary made by the PV-Model is not strong enough. We prove that, even under this assumption, it is *impossible* to achieve reader authentication and simultaneously *any* notion of privacy that allows tag corruption. This implies that the PV-Model cannot provide privacy along with mutual authentication without relying on tamper-proof hardware, which is unrealistic for low-cost RFID tags. Consequently, two of the three schemes presented by Paise and Vaudenay [150] do not satisfy their claimed properties. Our second contribution is to show that even under the strong assumption that the temporary tag state is not subject to tag corruption attacks, some privacy notions still remain impossible in the PV-Model. This implies that the third protocol by Paise and Vaudenay [150] has another conceptually different weakness.

Although our results are shown on the privacy model by Paise and Vaudenay, we believe that our work is helpful for developing a mature security and privacy model for RFID systems that fulfills the sophisticated requirements of real-life applications. So far, our results led to the refinement of the Paise and Vaudenay RFID security and privacy model [191] and they were considered in the development of improved security and privacy models for RFID systems [84, 53].

4.2 The Paise-Vaudenay RFID Security and Privacy Model

In this section, we recall the RFID security and privacy model by Paise and Vaudenay (PV-Model) [150] that refines the model by Vaudenay [190]. We give a more formal specification of this model, which is one of the most comprehensive RFID privacy and security models up to date.

4.2.1 System Model

The PV-Model considers RFID systems that consist of a single operator \mathcal{I} , a single reader \mathcal{R} and a polynomial number of tags \mathcal{T} . Note that the PV-Model does not explicitly define an entity that corresponds to the operator \mathcal{I} but implies the existence of such an entity. \mathcal{R} is assumed to be capable of performing public-key cryptography and of handling multiple instances of the mutual authentication protocol with different tags in parallel. Each tag \mathcal{T} is a passive device, i.e., it does not have its own power supply but is powered by the electromagnetic field of \mathcal{R} . Hence, \mathcal{T} cannot initiate communication, has a narrow communication range (i.e., a few centimeters to meters) and erases its temporary state (i.e., all session-specific information and randomness) after it gets out of the reading range of \mathcal{R} . Each \mathcal{T} is assumed to be capable of computing basic cryptographic functions like hashing, random number generation and symmetric-key encryption. Paise and Vaudenay [190, 150] also use public-key encryption, although it exceeds the capabilities of most currently available RFID tags [9, 141].

Security and privacy objectives. The main security objective of the PV-Model is mutual authentication. More precisely, \mathcal{R} should only accept legitimate tags and must be able to identify them, while each legitimate tag \mathcal{T} should only accept \mathcal{R} . Availability and protection against cloning are not captured by the PV-Model. The privacy objectives are anonymity and unlinkability.

Definitions. The operator \mathcal{I} sets up the reader \mathcal{R} and all tags \mathcal{T} . Hence, there are two setup algorithms where \mathcal{R} and \mathcal{T} are initialized and their system parameters (e.g., keys) are generated and defined. A protocol between \mathcal{T} and \mathcal{R} covers mutual authentication.

Definition 4.1 (RFID System [150]). *An RFID system is a tuple of probabilistic polynomial time algorithms $(\mathcal{R}, \mathcal{T}, \text{SetupReader}, \text{SetupTag}, \text{Auth})$ that are defined as follows:*

$\text{SetupReader}(1^l) \rightarrow (sk_{\mathcal{R}}, pk_{\mathcal{R}}, \text{DB})$ On input of a security parameter l , this algorithm creates the public parameters $pk_{\mathcal{R}}$ that are known to all entities. Moreover, it creates the secret parameters $sk_{\mathcal{R}}$ and a database DB that can only be accessed by \mathcal{R} .

$\text{SetupTag}_{pk_{\mathcal{R}}}(ID) \rightarrow (K, S)$ This function uses $pk_{\mathcal{R}}$ to generate a tag secret K and tag state S , initializes the tag \mathcal{T} with S and stores (ID, K) in DB .

$\text{Auth}[\mathcal{T} : S; \mathcal{R} : sk_{\mathcal{R}}, \text{DB}; * : pk_{\mathcal{R}}] \rightarrow [\mathcal{T} : out_{\mathcal{T}}; \mathcal{R} : out_{\mathcal{R}}]$ This is an interactive protocol between \mathcal{T} and \mathcal{R} . \mathcal{T} takes as input its current state S while \mathcal{R} has input $sk_{\mathcal{R}}$ and DB . The common input to all parties is $pk_{\mathcal{R}}$. After the protocol terminates, \mathcal{R} returns either the identity ID of \mathcal{T} or \perp to indicate that \mathcal{T} is not a legitimate tag. \mathcal{T} returns either 1 to indicate that \mathcal{R} is legitimate reader or \perp otherwise.²

Definition 4.2 (Correctness [190, 150]). An RFID system (Definition 4.1) is correct if $\forall l, \forall (sk_{\mathcal{R}}, pk_{\mathcal{R}}, \text{DB}) \in [\text{SetupReader}(1^l)]$ and $\forall (K, S) \in [\text{SetupTag}_{pk_{\mathcal{R}}}(ID)]$ with overwhelming probability $\text{Auth}[\mathcal{T} : S; \mathcal{R} : sk_{\mathcal{R}}, \text{DB}; * : pk_{\mathcal{R}}] \rightarrow [\mathcal{T} : 1; \mathcal{R} : ID]$.

4.2.2 Trust and Adversary Model

In the PV-Model, the issuer \mathcal{I} , the backend system \mathcal{D} (typically containing a database DB) and the readers are assumed to be trusted whereas a tag \mathcal{T} can be compromised. All readers and \mathcal{D} are subsumed to *one single* reader entity \mathcal{R} that cannot be corrupted. This implies that all readers are assumed to be tamper-resistant devices that have a permanent online connection to \mathcal{D} .³ The PV-Model defines privacy and security as security experiments, where a probabilistic polynomial time adversary \mathcal{A} can interact with a set of oracles that model the capabilities of \mathcal{A} . These oracles are:

$\text{CreateTag}^b(ID)$ Allows \mathcal{A} to set up a tag \mathcal{T} with identifier ID by internally calling $\text{SetupTag}_{pk_{\mathcal{R}}}(ID)$ to create (K, S) for \mathcal{T} . If input $b = 1$, then (ID, K) is added to DB . If $b = 0$, then (ID, K) is *not* added to DB .

² A *false negative* occurs when \mathcal{T} is legitimate but $out_{\mathcal{R}} = \perp$. A *false positive* happens if \mathcal{T} is not legitimate and $out_{\mathcal{R}} \neq \perp$. An *incorrect identification* occurs if the tag \mathcal{T} with identifier ID is legitimate but $out_{\mathcal{R}} \notin \{ID, \perp\}$

³ Depending on the use case, this assumption can be problematic in practice, e.g., for mobile readers that usually have only a sporadic or no online connection and that are subject to a variety of soft- and hardware attacks.

DrawTag(δ) $\rightarrow (vtag_1, b_1, \dots, vtag_n, b_n)$ Initially, \mathcal{A} cannot interact with any tag but must query **DrawTag** to get access to a set of tags chosen according to a probability distribution δ . \mathcal{A} knows the tags it can interact with by some temporary tag identifiers $vtag_1, \dots, vtag_n$. **DrawTag** manages a secret look-up table Γ that keeps track of the real tag identifier ID_i associated with each temporary tag identifier $vtag_i$, i.e., $\Gamma[vtag_i] = ID_i$. Moreover, **DrawTag** also provides \mathcal{A} with information on whether the tags are legitimate ($b_i = 1$) or not ($b_i = 0$).

FreeTag($vtag$) Makes tag $vtag$ inaccessible to \mathcal{A} such that \mathcal{A} cannot interact with $vtag$ until it is made accessible again under a new temporary identifier $vtag'$ by another **DrawTag** query.

LaunchIdent(π) $\rightarrow \pi$ Makes \mathcal{R} to start a new instance π of the **Auth** protocol.

SendReader(m, π) $\rightarrow m'$ Sends a message m to instance π of the **Auth** protocol that is running on \mathcal{R} . \mathcal{R} interprets m as a protocol message of instance π of the **Auth** protocol and responds with a message m' .

SendTag($m, vtag$) $\rightarrow m'$ Sends a message m to tag $vtag$, which interprets m as a protocol message of the **Auth** protocol and responds with a message m' .

Result(π) Returns 1 if instance π of the **Auth** protocol has been completed and the tag \mathcal{T} that participated in instance π has been accepted by \mathcal{R} . Otherwise **Result** returns 0.

CorruptTag($vtag$) $\rightarrow S$ Returns the current state S (i.e., all information stored in the memory) of the tag $vtag$ to \mathcal{A} .

The PV-Model distinguishes eight adversary classes, which differ in (1) their ability to corrupt tags and (2) the availability of auxiliary information, i.e., the ability to access the **CorruptTag** and **Result** oracle, respectively.

Definition 4.3 (Adversary Classes [150]). *An adversary is a probabilistic polynomial time algorithm that has arbitrary access to all oracles described in Section 4.2.2. Weak adversaries cannot access the **CorruptTag** oracle. Forward adversaries cannot query any other oracle than **CorruptTag** after they made the first **CorruptTag** query. Destructive adversaries cannot query any oracle for $vtag$ again after they made a **CorruptTag**($vtag$) query. Strong adversaries have no restrictions on the use of the **CorruptTag** oracle. Narrow adversaries cannot access the **Result** oracle.*

Observe that weak adversaries cannot corrupt tags and are limited to active attacks on the protocols. This assumes that corruption of tags is infeasible (e.g., due to tamper-resistant hardware), which is clearly not the case for low-cost RFID tags. Forward adversaries cannot interact with the RFID system (i.e., all the oracles described above) any longer after corrupting any of the tags for the first time but they can still make **CorruptTag** queries to all other tags. This models the case where the secrets of the tags become known when the life of the system is over. Destructive adversaries can never use a tag again after it has been corrupted but can still query all oracles for any of the remaining non-corrupted tags. This assumes that the tags are destroyed when they are corrupted (e.g., due to tamper-evident hardware). Strong adversaries have full access to all of the oracles at any time. Narrow adversaries cannot obtain auxiliary information, i.e., on whether a tag is legitimate or not. This may be the case in applications where the result of the identification protocol cannot be observed by the adversary. Therefore, a narrow adversary cannot query the **Result** oracle and is not given the values (b_1, \dots, b_n) from the **DrawTag** oracle, which both are the only sources of auxiliary information.

Tag corruption aspects. Depending on the concrete scenario, the full state including the temporary tag state is disclosed under tag corruption. In general, any concrete scenario will range between the following two extremes: (1) corruption discloses the full temporary tag state or (2) corruption does not disclose any information on the temporary tag state. In Section 4.3, we will prove that in both cases some privacy notions are impossible to achieve in the PV-Model. Thus, *independently* of any possible interpretation of tag corruption, impossibility results exist that contradict the claims in the PV-Model [150].

4.2.3 Security Definition

The security definition of the PV-Model focuses on attacks where the adversary aims to impersonate or forge a legitimate tag \mathcal{T} or the reader \mathcal{R} . It does *not* capture availability and security against cloning.

Tag authentication. The definition of tag authentication is based on a security experiment $\mathbf{Exp}_{\mathcal{A}_{\text{sec}}}^{\mathcal{T}\text{-aut}}$ where a strong adversary \mathcal{A}_{sec} (Definition 4.3) must make \mathcal{R} to identify some tag \mathcal{T} in some instance π of the Auth protocol. To exclude trivial attacks (e.g.,

relay attacks), \mathcal{A}_{sec} is not allowed to simply forward all the messages from \mathcal{T} to \mathcal{R} in instance π nor to corrupt \mathcal{T} . This means that at least some of the protocol messages that made \mathcal{R} to return ID must have been computed by \mathcal{A}_{sec} without knowing the secrets of \mathcal{T} . With $\mathbf{Exp}_{\mathcal{A}_{\text{sec}}}^{\mathcal{T}\text{-aut}} = 1$ we denote the case where \mathcal{A}_{sec} wins the security experiment.

Definition 4.4 (Tag Authentication [150]). *An RFID system (Definition 4.1) achieves tag authentication if for every strong adversary \mathcal{A}_{sec} (Definition 4.3) $\Pr[\mathbf{Exp}_{\mathcal{A}_{\text{sec}}}^{\mathcal{T}\text{-aut}} = 1]$ is negligible in the security parameter l .*

Reader authentication. The definition of reader authentication is based on a security experiment $\mathbf{Exp}_{\mathcal{A}_{\text{sec}}}^{\mathcal{R}\text{-aut}}$ where a strong adversary \mathcal{A}_{sec} (Definition 4.3) must successfully impersonate \mathcal{R} to a legitimate tag \mathcal{T} . Also here, to exclude trivial attacks, \mathcal{A}_{sec} must achieve this without simply forwarding the protocol messages from \mathcal{R} to \mathcal{T} . This means that \mathcal{A}_{sec} must have computed at least some of the protocol messages that made \mathcal{T} to return 1. With $\mathbf{Exp}_{\mathcal{A}_{\text{sec}}}^{\mathcal{R}\text{-aut}} = 1$ we denote the case where \mathcal{A}_{sec} wins the security experiment.

Definition 4.5 (Reader Authentication [150]). *An RFID system (Definition 4.1) achieves reader authentication if for every strong adversary \mathcal{A}_{sec} (Definition 4.3) $\Pr[\mathbf{Exp}_{\mathcal{A}_{\text{sec}}}^{\mathcal{R}\text{-aut}} = 1]$ is negligible in the security parameter l .*

Note that both tag and reader authentication are critical properties that must be preserved even against strong adversaries.

4.2.4 Privacy Definition

The privacy definition of the PV-Model is very flexible and, dependent on the adversary class (Definition 4.3), it covers different notions of privacy. It captures anonymity and unlinkability and focuses on the privacy leakage of the communication of tags with the reader. It is based on the existence of a simulator \mathcal{B} , called *blinder*, that can simulate the reader \mathcal{R} and any tag \mathcal{T} without knowing their secrets such that the adversary \mathcal{A}_{prv} cannot distinguish whether it is interacting with the real or the simulated RFID system. The rationale behind this simulation-based definition is that the communication of \mathcal{T} and \mathcal{R} does not leak any information about \mathcal{T} . Hence, everything \mathcal{A}_{prv} observes from the interaction with \mathcal{T} and \mathcal{R} appears to be independent of \mathcal{T} and consequently, \mathcal{A}_{prv} cannot distinguish different tags based on their communication.

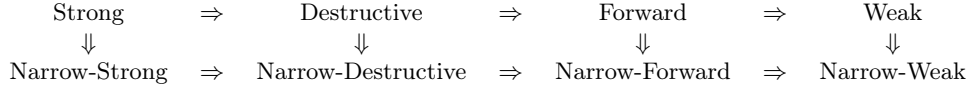


Figure 4.1: Privacy Notions Defined in the PV-Model and their Relations

This privacy definition can be formalized by the following experiment $\mathbf{Exp}_{\mathcal{A}_{\text{prv}}}^{\text{prv-}b} = b'$: Let \mathcal{A}_{prv} be an adversary according to Definition 4.3, l be the security parameter and $b \xleftarrow{\$} \{0, 1\}$. In the first phase of the experiment, \mathcal{R} is initialized with $(sk_{\mathcal{R}}, pk_{\mathcal{R}}, \text{DB}) \leftarrow \text{SetupReader}(1^l)$. The public key $pk_{\mathcal{R}}$ is given to \mathcal{A}_{prv} and \mathcal{B} . Then, \mathcal{A}_{prv} is allowed to arbitrarily interact with all oracles defined in Section 4.2.2. Hereby, \mathcal{A}_{prv} is subject to the restrictions of its corresponding adversary class (Definition 4.3). If $b = 1$, all queries to the **LaunchIdent**, **SendReader**, **SendTag** and the **Result** oracles are redirected to and answered by \mathcal{B} . Hereby, \mathcal{B} can observe all queries \mathcal{A}_{prv} makes to all other oracles that are not simulated by \mathcal{B} and the corresponding responses (“ \mathcal{B} sees what \mathcal{A}_{prv} sees”). After a polynomial number of oracle queries, the second phase of the experiment starts. In this second stage, \mathcal{A}_{prv} cannot interact with the oracles but is given the secret table Γ of the **DrawTag** oracle. Finally, \mathcal{A}_{prv} returns a bit b' .

Definition 4.6 (Privacy [190]). *Let C be one of the adversary classes according to Definition 4.3. An RFID system (Definition 4.1) is C -private if for every adversary \mathcal{A}_{prv} of C there exists a probabilistic polynomial time algorithm \mathcal{B} (blinder) such that the advantage*

$$\mathbf{Adv}_{\mathcal{A}_{\text{prv}}}^{\text{prv}} = |\Pr[\mathbf{Exp}_{\mathcal{A}_{\text{prv}}}^{\text{prv-}0} = 1] - \Pr[\mathbf{Exp}_{\mathcal{A}_{\text{prv}}}^{\text{prv-}1} = 1]|$$

*of \mathcal{A}_{prv} is negligible. \mathcal{B} simulates the **LaunchIdent**, **SendReader**, **SendTag** and the **Result** oracles to \mathcal{A}_{prv} without having access to $sk_{\mathcal{R}}$ and DB . Hereby, all oracle queries \mathcal{A}_{prv} makes and their corresponding responses are also sent to \mathcal{B} .*

All privacy notions defined in the PV-Model are summarized in Figure 4.1, which also shows their relations. It has been shown that strong privacy is impossible [190] while the technical feasibility of destructive privacy was an open problem. In Section 9.3, we present a destructive-private tag authentication protocol in a mild variant of Vaudenay’s RFID model [190].

4.3 The Paise-Vaudenay Model Revisited

4.3.1 Corruption with Full State Disclosure

We now point out a subtle weakness of the PV-Model. Specifically, we show that in the PV-Model it is *impossible* to achieve *any* notion of privacy simultaneously with reader authentication (under full state disclosure) except for the weak and narrow-weak privacy notions. As a consequence, two of the protocols given in [150] do not achieve their claimed privacy goals.

We stress that this impossibility result is due to the fact that, according to the formal definitions of the PV-Model, the adversary can obtain the *full* state including the temporary memory of a tag by corrupting the tag *while* it is executing a protocol with the reader. Such attacks are a serious threat in practice, in particular to low-cost RFID tags and hence must be formally considered. Although Paise and Vaudenay [150] informally discuss an issue related to tag corruption during protocol execution, we show that such attacks are *not* adequately captured by the formal definitions of the PV-Model. Hence, the only achievable privacy notions are those where the adversary is not allowed to corrupt tags at all. Since in practice tag corruption is realistic, this implies that using the PV-Model is not helpful when reader authentication and a reasonable notion of privacy are needed.

Impossibility of narrow-forward privacy. To prove our first impossibility result, we need the following lemma, which we will prove in detail further below:

Lemma 4.1 (Blinder Violates Reader Authentication). *If there is a blinder \mathcal{B} for every narrow-forward adversary \mathcal{A}_{prv} such that $\text{Adv}_{\mathcal{A}_{\text{prv}}}^{\text{prv}}$ is negligible in the security parameter l (Definition 4.6), then \mathcal{B} can be used to construct an adversary $\mathcal{A}_{\text{sec}}^{\mathcal{B}}$ such that $\Pr[\text{Exp}_{\mathcal{A}_{\text{sec}}^{\mathcal{B}}}^{\mathcal{R}\text{-aut}} = 1]$ is non-negligible in l (Definition 4.5).*

Based on this lemma, we set up the following theorem, which we need later to prove our main impossibility result:

Theorem 4.1 (Narrow-Forward Privacy Contradicts Reader Authentication). *There is no RFID system (Definition 4.1) that achieves both reader authentication (Definition 4.5) and narrow-forward privacy (Definition 4.6) under full tag state disclosure.*

Proof of Theorem 4.1. Let \mathcal{A}_{prv} be a narrow-forward adversary (Definition 4.3). Definition 4.6 requires the existence of a blinder \mathcal{B} such that \mathcal{A}_{prv} cannot distinguish \mathcal{B} from the real oracles. From Lemma 4.1 it follows that \mathcal{B} can be used to impersonate \mathcal{R} to any legitimate tag \mathcal{T} with non-negligible probability. Hence, the existence of \mathcal{B} contradicts reader authentication (Definition 4.5). \square

Proof of Lemma 4.1. First, we show how to construct $\mathcal{A}_{\text{sec}}^{\mathcal{B}}$ from \mathcal{B} . Second, we prove that $\mathcal{A}_{\text{sec}}^{\mathcal{B}}$ violates reader authentication (Definition 4.5) if \mathcal{B} is such that $\text{Adv}_{\mathcal{A}_{\text{prv}}}^{\text{prv}}$ is negligible for every narrow-forward \mathcal{A}_{prv} (Definition 4.3).

Let $q_{\mathcal{R}} \in \mathbb{N}$ with $q_{\mathcal{R}} > 0$ be the (expected) number of **SendReader** queries as specified by the **Auth** protocol and let $S_i^{\mathcal{R}}$ be the state of \mathcal{R} after processing the i -th **SendReader** query. The initial reader state $S_0^{\mathcal{R}}$ includes the public key $pk_{\mathcal{R}}$ and the secret key $sk_{\mathcal{R}}$ of \mathcal{R} as well as a pointer to the credentials database DB. Note that during the processing of a **SendReader** query, \mathcal{R} can update DB. \mathcal{R} can be considered as a tuple of algorithms $(\mathcal{R}_{\pi}^{(1)}, \dots, \mathcal{R}_{\pi}^{(q_{\mathcal{R}})})$, where $\mathcal{R}_{\pi}^{(i)}$ represents the computation performed by \mathcal{R} when processing the i -th **SendReader** query in instance π of the **Auth** protocol. Formally: $(S_1^{\mathcal{R}}, m_1) \leftarrow \mathcal{R}_{\pi}^{(0)}(S_0^{\mathcal{R}})$ and $(S_{i+1}^{\mathcal{R}}, m_{2i+1}) \leftarrow \mathcal{R}_{\pi}^{(i)}(S_i^{\mathcal{R}}, m_{2i})$ for $1 \leq i < q_{\mathcal{R}}$. Since tags are passive devices that cannot initiate communication \mathcal{R} must send the first protocol message. Thus, \mathcal{R} generates all protocol messages with odd indices whereas the tag \mathcal{T} generates all messages with even indices. In case the **Auth** protocol specifies that \mathcal{T} sends the last protocol message, then $m_{2q_{\mathcal{R}}-1}$ is the empty string.

Let $q_{\mathcal{T}} \in \mathbb{N}$ with $q_{\mathcal{T}} > 0$ be the (expected) number of **SendTag** queries as specified by the **Auth** protocol and let $S_i^{\mathcal{T}}$ be the state of \mathcal{T} after processing the i -th **SendTag** query. \mathcal{T} can be represented as a tuple of algorithms $(\mathcal{T}^{(1)}, \dots, \mathcal{T}^{(q_{\mathcal{T}})})$ where $\mathcal{T}^{(i)}$ means the computation performed by \mathcal{T} when processing the i -th **SendTag** query in an instance of the **Auth** protocol that involves \mathcal{T} . Formally: $(S_{i+1}^{\mathcal{T}}, m_{2i}) \leftarrow \mathcal{T}^{(i)}(S_i^{\mathcal{T}}, m_{2i-1})$ for $1 \leq i \leq q_{\mathcal{T}}$. Note that $m_{2q_{\mathcal{T}}}$ is the empty string if **Auth** specifies that \mathcal{R} must send the last protocol message.

The idea of $\mathcal{A}_{\text{sec}}^{\mathcal{B}}$ is to internally use \mathcal{B} as a black-box to simulate the final protocol message of \mathcal{R} that makes each legitimate tag \mathcal{T} accept $\mathcal{A}_{\text{sec}}^{\mathcal{B}}$ as \mathcal{R} . The construction of $\mathcal{A}_{\text{sec}}^{\mathcal{B}}$ is shown in Algorithm 1. First, $\mathcal{A}_{\text{sec}}^{\mathcal{B}}$ creates a legitimate tag \mathcal{T} (Step 1) and makes it accessible (Step 2). Both steps are also shown to \mathcal{B} , which expects to observe all oracle queries. Then, $\mathcal{A}_{\text{sec}}^{\mathcal{B}}$ makes \mathcal{B} to start a new instance π of the **Auth** protocol with \mathcal{T} (Step 3) and obtains the first protocol message m_1 generated by \mathcal{B} (Step 4). Now, $\mathcal{A}_{\text{sec}}^{\mathcal{B}}$

Algorithm 1 Adversary $\mathcal{A}_{\text{sec}}^{\mathcal{B}}$ violating reader authentication

```

1: CreateTag( $ID$ )
2:  $vtag \leftarrow \text{DrawTag}(\Pr[ID] = 1)$ 
3:  $\pi \leftarrow \text{LaunchIdent}()$  ▷ simulated by  $\mathcal{B}$ 
4:  $m_1 \leftarrow \text{SendReader}(-, \pi)$  ▷ simulated by  $\mathcal{B}$ 
5:  $i \leftarrow 1$ 
6: while  $i < q_{\mathcal{R}}$  do
7:   if  $i \leq q_{\mathcal{T}}$  then  $m_{2i} \leftarrow \text{SendTag}(m_{2i-1}, vtag)$  ▷ simulated by  $\mathcal{B}$ 
8:   end if
9:    $m_{2i+1} \leftarrow \text{SendReader}(m_{2i}, \pi)$  ▷ simulated by  $\mathcal{B}$ 
10:   $i \leftarrow i + 1$ 
11: end while
12:  $out_{\mathcal{T}} \leftarrow \text{SendTag}(m_{2q_{\mathcal{R}}-1}, vtag)$  ▷ computed by  $\mathcal{T}$ 
    
```

internally runs \mathcal{B} that simulates both \mathcal{T} and \mathcal{R} until \mathcal{B} returns the final reader message $m_{2q_{\mathcal{R}}-1}$ (Steps 5–11). Finally, $\mathcal{A}_{\text{sec}}^{\mathcal{B}}$ sends $m_{2q_{\mathcal{R}}-1}$ to the real tag \mathcal{T} (Step 12). $\mathcal{A}_{\text{sec}}^{\mathcal{B}}$ succeeds if \mathcal{T} accepts \mathcal{B} as \mathcal{R} . Formally, this means that:

$$\Pr \left[\text{Exp}_{\mathcal{A}_{\text{sec}}^{\mathcal{B}}}^{\mathcal{R}\text{-aut}} = 1 \right] = \Pr \left[\text{Auth}[\mathcal{T} : S_0^{\mathcal{T}}; \mathcal{A}_{\text{sec}}^{\mathcal{B}} : -; * : pk_{\mathcal{R}}] \rightarrow [\mathcal{T} : 1; \mathcal{A}_{\text{sec}}^{\mathcal{B}} : \cdot] \right] \quad (4.1)$$

We stress that this indeed is a valid attack with regard to Definition 4.5 since \mathcal{A}_{sec} does not just forward the protocol messages between \mathcal{R} and \mathcal{T} .

Next, we show that narrow-forward privacy (Definition 4.6) ensures that $\mathcal{A}_{\text{sec}}^{\mathcal{B}}$ succeeds with non-negligible probability, i.e., that Equation 4.1 is non-negligible. Note that in case Equation 4.1 is negligible, this implies that with non-negligible probability p_{\perp} message $m_{2q_{\mathcal{R}}-1}$ generated by \mathcal{B} makes \mathcal{T} to return $out_{\mathcal{T}} = \perp$. In the following, we show that if p_{\perp} is non-negligible, then there is a narrow-forward adversary \mathcal{A}_{prv} that has non-negligible advantage $\text{Adv}_{\mathcal{A}_{\text{prv}}}^{\text{prv}}$ to distinguish \mathcal{B} from the real oracles, which contradicts narrow-forward privacy (Definition 4.6). The construction of \mathcal{A}_{prv} is shown in Algorithm 2. First, \mathcal{A}_{prv} creates a legitimate tag \mathcal{T} (Step 1) and makes it accessible (Step 2). Then, \mathcal{A}_{prv} makes \mathcal{R} to start a new instance π of the **Auth** protocol with \mathcal{T} (Step 3) and obtains the first protocol message m_1 from \mathcal{R} (Step 4). Now, \mathcal{A}_{prv} eavesdrops on the execution of the **Auth** protocol up to the point *after* \mathcal{R} has sent its last protocol message $m_{2q_{\mathcal{R}}-1}$ (Steps 5–11) and corrupts \mathcal{T} just *before* \mathcal{T} received $m_{2q_{\mathcal{R}}-1}$ (Step 12). Next, \mathcal{A}_{prv} performs the computation \mathcal{T} would have done on receipt of $m_{2q_{\mathcal{R}}-1}$ (Step 13).

Algorithm 2 Narrow-forward adversary \mathcal{A}_{prv}

```

1: CreateTag( $ID$ )
2:  $vtag \leftarrow \text{DrawTag}(\text{Pr}[ID] = 1)$ 
3:  $\pi \leftarrow \text{LaunchIdent}()$ 
4:  $m_1 \leftarrow \text{SendReader}(-, \pi)$ 
5:  $i \leftarrow 1$ 
6: while  $i < q_{\mathcal{R}}$  do
7:   if  $i \leq q_{\mathcal{T}}$  then  $m_{2i} \leftarrow \text{SendTag}(m_{2i-1}, vtag)$ 
8:   end if
9:    $m_{2i+1} \leftarrow \text{SendReader}(m_{2i}, \pi)$ 
10:   $i \leftarrow i + 1$ 
11: end while
12:  $S_{q_{\mathcal{R}}}^{\mathcal{T}} \leftarrow \text{CorruptTag}(vtag)$ 
13:  $out_{\mathcal{T}} \leftarrow \mathcal{T}^{(q_{\mathcal{R}})}(S_{q_{\mathcal{R}}}^{\mathcal{T}}, m_{2q_{\mathcal{R}}-1})$ 
14: if  $out_{\mathcal{T}} = 1$  then return 0
15: else return 1
16: end if

```

If this computation results in $out_{\mathcal{T}} = 1$, \mathcal{A}_{prv} returns 0 to indicate that it interacted with the real oracles (Step 14). Otherwise, \mathcal{A}_{prv} indicates the presence of \mathcal{B} by returning 1 (Step 15). Note that \mathcal{A}_{prv} indeed is a narrow-forward adversary (Definition 4.3) since \mathcal{A}_{prv} never queries **Result** and none of the oracles in Section 4.2.2 after corrupting \mathcal{T} .

Next, we show that \mathcal{A}_{prv} has non-negligible advantage $\mathbf{Adv}_{\mathcal{A}_{\text{prv}}}^{\text{prv}}$ if p_{\perp} is non-negligible. Therefore, we first consider the case where \mathcal{A}_{prv} interacts with the real oracles. Since \mathcal{T} is legitimate, it follows from correctness (Definition 4.2) that $out_{\mathcal{T}} = 1$ with overwhelming probability p_1 . Hence, $\Pr[\mathbf{Exp}_{\mathcal{A}_{\text{prv}}}^{\text{prv-0}} = 1] = 1 - p_1$ is negligible. Now, consider the case where \mathcal{A}_{prv} interacts with \mathcal{B} . Note that by the contradicting hypothesis, \mathcal{B} generates a protocol message $m_{2q_{\mathcal{R}}-1}$ that makes \mathcal{T} to return $out_{\mathcal{T}} = \perp$ with non-negligible probability p_{\perp} . Thus, we have $\Pr[\mathbf{Exp}_{\mathcal{A}_{\text{prv}}}^{\text{prv-1}} = 1] = p_{\perp}$. Hence, it follows that $\mathbf{Adv}_{\mathcal{A}_{\text{prv}}}^{\text{prv}} = |1 - p_1 - p_{\perp}|$. Note that due to correctness p_1 is overwhelming and by assumption p_{\perp} is non-negligible. Hence, $\mathbf{Adv}_{\mathcal{A}_{\text{prv}}}^{\text{prv}}$ is non-negligible, which contradicts narrow-forward privacy (Definition 4.6). In turn, this means that narrow-forward privacy ensures that Equation 4.1 is non-negligible, which finishes the proof. \square

Since the impossibility of narrow-forward privacy (Theorem 4.1), implies the impossibility of all other stronger privacy notions (cf. Figure 4.1), we have the following corollary,

which corresponds to the first main claim of this section:

Corollary 4.1 (No Private Mutual Authentication Under Temporary State Disclosure). *In the PV-Model there is no RFID system (Definition 4.1) that achieves both reader authentication (Definition 4.5) and any privacy notion that is different from weak and narrow-weak privacy (Definition 4.6) under full state disclosure.*

4.3.2 Corruption without Temporary State Disclosure

Our first impossibility result shows that the PV-Model requires further assumptions to evaluate the privacy properties of RFID systems where tag corruption is of concern. A natural question therefore is, whether one can achieve mutual authentication along with some form of privacy if the temporary tag state is *not* disclosed. Hence, in this section we consider the case where corruption reveals *only* the persistent tag state but *no* information on the temporary tag state.

The attack and the impossibility result shown in Section 4.3 critically use the fact that in the PV-Model the adversary \mathcal{A}_{prv} can learn the temporary state of a tag during the Auth protocol. This allows \mathcal{A}_{prv} to verify the response of \mathcal{R} (that may have been simulated by \mathcal{B}) and hence, due to reader authentication (Definition 4.5), \mathcal{A}_{prv} can distinguish with non-negligible advantage between the real oracles and \mathcal{B} . However, if \mathcal{A}_{prv} cannot obtain temporary tag states, it cannot perform this verification. Hence, the impossibility result we proved in Section 4.3 does not necessarily hold if the temporary state is safe to corruption.

Impossibility of narrow-strong privacy. We now show our second impossibility result. In the PV-Model, it is *impossible* to achieve narrow-strong privacy along with reader authentication. This means that, even in case the adversary cannot obtain the temporary tag state, the most challenging privacy notion defined in the PV-Model [150] (narrow-strong privacy) still remains unachievable. This implies a conceptually different weakness of the claimed narrow-strong private protocol in [150].

Theorem 4.2 (Narrow-Strong Privacy Contradicts Reader Authentication). *In the PV-Model there is no RFID system (Definition 4.1) that fulfills both reader authentication (Definition 4.5) and narrow-strong privacy (Definition 4.6).*

Algorithm 3 Adversary $\mathcal{A}_{\text{prv}}^{\mathcal{B}}$ violating reader authentication

```

1: CreateTag( $ID$ )
2:  $vtag \leftarrow \text{DrawTag}(\text{Pr}[ID] = 1)$ 
3:  $S_0^{\mathcal{T}} \leftarrow \text{CorruptTag}(vtag)$ 
4:  $\pi \leftarrow \text{LaunchIdent}()$  ▷ simulated by  $\mathcal{B}$ 
5:  $m_1 \leftarrow \text{SendReader}(-, \pi)$  ▷ simulated by  $\mathcal{B}$ 
6:  $i \leftarrow 1$ 
7: while  $i < q_{\mathcal{R}}$  do
8:   if  $i \leq q_{\mathcal{T}}$  then  $m_{2i} \leftarrow \text{SendTag}(m_{2i-1}, vtag)$ 
9:   end if
10:   $m_{2i+1} \leftarrow \text{SendReader}(m_{2i}, \pi)$  ▷ simulated by  $\mathcal{B}$ 
11:   $i \leftarrow i + 1$ 
12: end while
13:  $out_{\mathcal{T}} \leftarrow \text{SendTag}(m_{2q_{\mathcal{R}}-1}, vtag)$  ▷ computed by  $\mathcal{T}$ 

```

Proof of Theorem 4.2. Narrow-strong privacy (Definition 4.6) requires the existence of a blinder \mathcal{B} that simulates the **LaunchIdent**, **SendReader** and **SendTag** oracles such that every narrow-strong adversary \mathcal{A}_{prv} has negligible advantage $\mathbf{Adv}_{\mathcal{A}_{\text{prv}}}^{\text{prv}}$ to distinguish \mathcal{B} from the real oracles. We show that \mathcal{B} can be used to construct an algorithm $\mathcal{A}_{\text{prv}}^{\mathcal{B}}$ that violates reader authentication (Definition 4.5).

The construction of $\mathcal{A}_{\text{prv}}^{\mathcal{B}}$ is as shown in Algorithm 3. First, $\mathcal{A}_{\text{prv}}^{\mathcal{B}}$ creates a legitimate tag \mathcal{T} (Step 1), makes it accessible (Step 2) and corrupts it (Step 3). These three steps are also shown to \mathcal{B} , which expects to observe all oracle queries. Then, $\mathcal{A}_{\text{prv}}^{\mathcal{B}}$ makes \mathcal{B} to start a new instance π of the **Auth** protocol with \mathcal{T} (Step 4) and obtains the first protocol message m_1 generated by \mathcal{B} (Step 5). Now, $\mathcal{A}_{\text{prv}}^{\mathcal{B}}$ internally runs \mathcal{B} that simulates $vtag$ and \mathcal{R} until \mathcal{B} returns the final reader message $m_{2q_{\mathcal{R}}-1}$ (Steps 6–12). Finally, $\mathcal{A}_{\text{prv}}^{\mathcal{B}}$ sends $m_{2q_{\mathcal{R}}-1}$ to the real tag \mathcal{T} (Step 13). $\mathcal{A}_{\text{prv}}^{\mathcal{B}}$ succeeds if \mathcal{T} accepts $m_{2q_{\mathcal{R}}-1}$ and returns $out_{\mathcal{T}} = 1$, which means that \mathcal{T} accepts \mathcal{B} as \mathcal{R} . Formally:

$$\Pr \left[\mathbf{Exp}_{\mathcal{A}_{\text{prv}}^{\mathcal{B}}}^{\mathcal{R}\text{-aut}} = 1 \right] = \Pr \left[\text{Auth}[\mathcal{T} : S_0^{\mathcal{T}}; \mathcal{A}_{\text{prv}}^{\mathcal{B}} : -; * : pk_{\mathcal{R}}] \rightarrow [\mathcal{T} : 1; \mathcal{A}_{\text{prv}}^{\mathcal{B}} : \cdot] \right] \quad (4.2)$$

We stress that this indeed is a valid attack with regard to Definition 4.5 since \mathcal{A}_{sec} does not just forward the protocol messages between \mathcal{R} and \mathcal{T} .

From reader authentication (Definition 4.5) it follows that Equation 4.2 must be negligible. However, this implies that with overwhelming probability \mathcal{B} generates at least one

Algorithm 4 Narrow-strong adversary \mathcal{A}_{prv}

```

1: CreateTag( $ID$ )
2:  $vtag \leftarrow \text{DrawTag}(\text{Pr}[ID] = 1)$ 
3:  $S_0^{\mathcal{T}} \leftarrow \text{CorruptTag}(vtag)$ 
4:  $\pi \leftarrow \text{LaunchIdent}()$ 
5:  $m_1 \leftarrow \text{SendReader}(-, \pi)$ 
6:  $t \in \{1, \dots, q_{\mathcal{T}}\}$ 
7:  $i \leftarrow 1$ 
8: while  $i < t$  do
9:    $(S_{i+1}^{\mathcal{T}}, m_{2i}) \leftarrow \mathcal{T}^{(i)}(S_i^{\mathcal{T}}, m_{2i-1})$ 
10:  if  $i < q_{\mathcal{R}}$  then  $m_{2i+1} \leftarrow \text{SendReader}(m_{2i}, \pi)$ 
11:  end if
12:   $i \leftarrow i + 1$ 
13: end while
14:  $out_{\mathcal{T}} \leftarrow \mathcal{T}^{(t)}(S_t^{\mathcal{T}}, m_{2t-1})$ 
15: if  $out_{\mathcal{T}} = 1$  then return 0
16: else return 1
17: end if
    
```

protocol message that makes \mathcal{T} to finally return $out_{\mathcal{T}} = \perp$. Let p_t be the probability that this is the case for message m_{2t-1} for some $t \in \{1, \dots, q_{\mathcal{T}}\}$. We now show a narrow-strong adversary \mathcal{A}_{prv} that succeeds with non-negligible advantage $\mathbf{Adv}_{\mathcal{A}_{\text{prv}}}^{\text{prv}}$ if p_t is non-negligible, which contradicts narrow-strong privacy (Definition 4.6). The construction of \mathcal{A}_{prv} is shown in Algorithm 4. First, \mathcal{A}_{prv} creates a legitimate tag \mathcal{T} (Step 1), makes it accessible (Step 2) and corrupts it (Step 3). Note that by a **CorruptTag** query, \mathcal{A}_{prv} only learns the persistent tag state $S_0^{\mathcal{T}}$ of \mathcal{T} . Then, \mathcal{A}_{prv} makes \mathcal{R} to start an instance π of the **Auth** protocol with \mathcal{T} (Step 4) and obtains the first protocol message m_1 from \mathcal{R} (Step 5). Now, \mathcal{A}_{prv} guesses t (Step 6) and simulates \mathcal{T} (using $S_0^{\mathcal{T}}$) in the **Auth** protocol up to the point where **SendReader** returns message m_{2t-1} (Steps 7–13). Next, \mathcal{A}_{prv} performs the computation \mathcal{T} would have done on receipt of message m_{2t-1} (Step 14). Finally, \mathcal{A}_{prv} returns either 0 to indicate that it interacted with the real oracles (Step 15) or 1 to indicate the presence of \mathcal{B} (Step 16).

Next, we show that \mathcal{A}_{prv} has non-negligible $\mathbf{Adv}_{\mathcal{A}_{\text{prv}}}^{\text{prv}}$ if p_{\perp} is non-negligible. Therefore, we first consider the case where \mathcal{A}_{prv} interacts with the real oracles. Since \mathcal{T} is legitimate, it follows from correctness (Definition 4.2) that $out_{\mathcal{T}} = 1$ with overwhelming probability p_1 . This means that $\Pr[\mathbf{Exp}_{\mathcal{A}_{\text{prv}}}^{\text{prv-0}} = 1] = 1 - p_1$ is negligible. Now, consider the

case where \mathcal{A}_{prv} interacts with \mathcal{B} . Note that by the contradicting hypothesis, with non-negligible probability p_t \mathcal{B} generates a message m_{2t-1} that makes \mathcal{T} to return $\text{out}_{\mathcal{T}} = \perp$. Moreover, \mathcal{A}_{prv} guesses t with probability of at least $1/q_{\mathcal{T}}$. Thus, we have $\Pr[\mathbf{Exp}_{\mathcal{A}_{\text{prv}}}^{\text{prv-1}} = 1] \geq \frac{p_t}{q_{\mathcal{T}}}$. Hence, it follows that $\mathbf{Adv}_{\mathcal{A}_{\text{prv}}}^{\text{prv}} \geq |1 - p_1 - \frac{p_t}{q_{\mathcal{T}}}|$. Note that due to correctness p_1 is overwhelming while p_t is non-negligible by assumption and $q_{\mathcal{T}}$ is polynomially bounded. Hence, $\mathbf{Adv}_{\mathcal{A}_{\text{prv}}}^{\text{prv}}$ is non-negligible, which contradicts narrow-strong privacy (Definition 4.6) and finishes the proof. \square

4.4 Conclusion

In this chapter, we revisited the security and privacy model for RFID systems proposed by Paise and Vaudenay (PV-Model) [150]. This model is very interesting since it covers many aspects of previous works, proposes a unified RFID security and privacy framework and is the basis of many subsequent works [137, 138, 32, 165, 164, 50, 49, 168, 167]. We proved several impossibility results that show that the formalization given in the PV-Model is too restrictive and fails in modelling real-life scenarios, where interesting privacy notions and reader authentication are intuitively achievable. These impossibility results seem to be related to the notion of the *blinder* on which the privacy definition of the PV-Model is based on. The blinder must simulate all tags and readers in the system while the adversary learns the state of the real tag when corrupting it. Since the state of the real tag is independent of the simulation by the blinder, the adversary can distinguish the blinder from the real system and hence, violate the privacy definition.

Our results led to the refinement of the PV-Model [191] and they were considered in the development of improved security and privacy models for RFID systems [84, 53]. Although our results are on the privacy model by Paise and Vaudenay, we believe that they are of general importance for developing a mature security and privacy model for RFID systems.

5 Anonymizer-enabled Security and Privacy for RFID

In this chapter, we present the first security and privacy framework for anonymizer-enabled RFID systems and two privacy-preserving RFID authentication schemes using anonymizers. Both schemes achieve several appealing features that were not simultaneously achieved by any previous proposal. The first scheme is very efficient for all involved entities, in particular for the tags which only have to perform minimal computations, achieves privacy under tag corruption and it is secure against impersonation attacks and forgeries even if the adversary can corrupt the anonymizers. The second scheme provides for the first time anonymity and untraceability of tags against readers as well as secure tag authentication against collusions of malicious readers and anonymizers using cost-efficient tags that cannot perform public-key cryptography (i.e., modular exponentiations).

Remark. The results presented in this chapter are due to the author of this work and the result of many intensive discussions with Frederik Armknecht (University of Mannheim, Germany), Liqun Chen (HP Labs, UK), Ahmad-Reza Sadeghi (TU Darmstadt, Germany) and Ivan Visconti (University of Salerno, Italy). Parts of this chapter have been published in [165, 164] and in [2].

5.1 Motivation and Contribution

A promising approach to enhance the privacy in RFID systems without lifting the computational requirements on tags are anonymizers (cf. Section 3.3.3). These are special devices that take off the computational workload (i.e., the public-key operations) from tags and enable privacy-preserving protocols with cost-efficient tags. Note that anonymizer-based RFID systems are *not* a straight-forward extension of a resource constrained RFID system to one with more capabilities. This is because an additional protocol is required between the tags and the anonymizers opening new attack surfaces that must be carefully considered. Indeed, to ensure availability, the protocol between the tag and the anonymizer must be secure against attacks where the adversary aims to manipulate the

tag (denial-of-service). Moreover, an anonymizer shall not be able to impersonate or to copy the tags it anonymizes since this would violate tag authentication.

There are different ways to realize anonymizers. One approach is to provide public anonymizers that are controlled by the operator of the RFID system or by one of several independent anonymizer service providers the user may choose from. Alternatively, each user may have his/her own personal anonymizer that could be implemented as a software running on the user's mobile phone or PDA¹, allowing a very cost-efficient implementation of anonymizers. The main advantage of anonymizer-enabled protocols is that they allow operators of RFID systems to enable privacy for the concerned users (who may buy their own personal anonymizer) with only minor extra costs.

However, as discussed in Section 3.3.3, current anonymizer-enabled solutions are vulnerable to impersonation and cloning attacks. Moreover, existing security and privacy models for RFID (cf. Section 3.2) do not include anonymizers, which play a critical role for going beyond the barrier of simultaneously achieving a strong privacy notion with protocols that are suitable for cost-efficient tags.

In this chapter, we investigate the use of anonymizers in RFID systems with both trusted and untrusted readers (trust models TM3 and TM2 in Section 3.1.3, respectively). In this context, we focus on tag authentication and do *not* consider mutual authentication. More detailed, the contribution of this chapter is as follows:

Anonymizers in RFID Systems with Trusted Readers. We introduce a security and privacy model for anonymizer-enabled RFID systems that builds on top of the RFID security and privacy framework by Vaudenay [190], which we call the Vaudenay-Model (V-Model).² We present a privacy-preserving RFID protocol that uses anonymizers and achieves the strongest achievable notion of privacy in this model (*narrow-strong privacy*) without requiring tags to perform expensive public-key operations (i.e., modular exponentiations), thus providing a satisfying notion of privacy for cost-efficient tags.

More detailed, we introduce a formal framework for privacy-preserving RFID systems

¹An increasing number of mobile phones and PDAs support the Near Field Communication (NFC) standard which enables them to communicate to RFID devices. Further, as discussed in Section 3.3.2, solutions using mobile computing devices as the only authentication token have several limitations: These devices may run out of power (which violates availability) and can be compromised by Trojans, which brings up new security challenges.

²Observe that, in contrast to the PV-Model [150] (cf. Section 4.2), the V-Model [190] considers only tag authentication (and no mutual authentication) and thus is not subject to the issues pointed out in Section 4.3.

that extends the V-Model to support anonymizers and at the same time is backwards-compatible to it. Given the granularity of the different security and privacy notions of the V-Model, our anonymizer-based model is the first universal security and privacy model for anonymizer-enabled RFID systems. Moreover, we propose a privacy-preserving RFID protocol that is secure and private in the anonymizer-enabled model. The protocol that we propose enjoys several appealing features which have not been simultaneously achieved by any previous proposal. Indeed, our protocol is very efficient for all involved entities, in particular for tags that only have to perform minimal computations. The protocol enjoys the strongest achievable privacy notion defined in the V-Model, which is narrow-strong privacy.³ Our protocol also provides forward privacy, which restricts the adversary's capability to corrupt tags but instead allows him to access auxiliary information. We finally stress that our protocol is provably secure against impersonation attacks and forgeries even if the adversary can corrupt the anonymizers. Therefore, we require the existence of (honest) anonymizers in the RFID system only to guarantee privacy (anonymity and untraceability) and availability of the tags. This assumption gracefully matches the realistic scenario where many anonymizers are spread in the system and the adversary can be successful in corrupting many of them with the purpose of violating the security (tag authentication) of the system. At the same time, privacy is guaranteed as long as tags are frequently anonymized by an uncorrupted anonymizer.

Anonymizers in RFID Systems with Untrusted Readers. We present an anonymous authentication scheme that enables RFID tags to authenticate to readers without disclosing the tag identity or any other information that allows tracing the tags. The properties of our scheme are very useful for a variety of access control systems, where it is sufficient or mandatory to verify the authenticity of a tag without inferring the tag identity to the reader. Our scheme adapts the anonymous authentication scheme by Chen et al. [42]⁴ for our purposes and uses anonymizers to ensure the anonymity and unlinkability of tags. This allows using cost-effective RFID tags that cannot perform public-key cryptography in an efficient and scalable way. Our solution provides (1) anonymity and

³Note that the impossibility of achieving strong privacy [190] trivially holds in our anonymizer-enabled model since any protocol in the anonymizer-enabled model also works in the V-Model by simply requiring that the anonymization protocol (i.e., the protocol run between tags and anonymizers) is played locally inside tags.

⁴Note that [42] has been retracted in 2011 due to flaws in the security proofs. However, the security analysis of our protocol is independent of the proofs in [42].

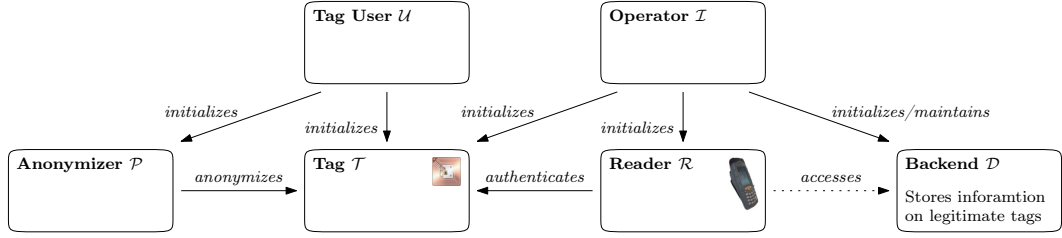


Figure 5.1: Anonymizer-Enabled RFID System

untraceability of tags against readers, (2) secure tag authentication even against collusions of malicious readers and anonymizers and (3) security against denial-of-service attacks on the protocol-level. This is a major improvement to existing RFID systems that usually require the strong assumption of trusted readers (cf. Section 3.2) and where a compromised reader usually has a severe impact on the security and privacy of all the tags in the system [12, 65, 139]. In contrast to existing solutions to anonymous tag authentication (cf. Section 3.3), our scheme matches the computational capabilities of common RFID tags.

5.2 Anonymizer-enabled RFID Systems

An anonymizer-enabled RFID system consists of (at least) a trusted tag issuer \mathcal{I} , a tag \mathcal{T} , a reader \mathcal{R} that may be connected to some backend infrastructure \mathcal{D} typically hosting a database DB and an anonymizer \mathcal{P} (cf. Figure 5.1). The issuer \mathcal{I} initializes the readers, the backend and the tags that can later be identified by all the readers in the system. The anonymizers are initialized by their respective owners (e.g., a privacy service provider or the tag user \mathcal{U}) and their task is to enforce the privacy goals of legitimate tags.

An authentication scheme using anonymizers consists of two protocols: the *tag authentication* and the *tag anonymization* protocol. The former is executed by \mathcal{R} and \mathcal{T} and allows \mathcal{R} to check whether \mathcal{T} is legitimate. The tag anonymization protocol ensures the privacy goals of the tag user \mathcal{U} by updating the authentication data (e.g., authentication secrets and certificates) of \mathcal{T} . Note that we do *not* assume that the tags can perform public-key cryptography since this exceeds the capabilities of most currently available RFID tags. However, \mathcal{T} is assumed to be capable of performing basic cryptographic operations such as random number generation, (lightweight) symmetric-key encryption and hashing.

5.3 Anonymizers in RFID Systems with Trusted Readers

We present an anonymous authentication scheme that enables RFID tags to authenticate to readers without disclosing their tag identity or any other information that allows tags to be traced. The scheme uses anonymizers to ensure anonymity and unlinkability of tags, which allows using cost-effective RFID tags that cannot perform public-key cryptography in an efficient and scalable way. Our solution provides (1) anonymity and untraceability of tags against readers, (2) secure tag authentication even against a collusion of malicious readers and anonymizers, and (3) security against denial-of-service attacks on the protocol-level.

Our scheme combines and extends some of the schemes proposed by Vaudenay [190] and employs anonymizers, which brings several improvements that are important for practical applications: The protocol achieves both narrow-strong and forward privacy, allows the tags to be verified in constant time and provides basic protection against denial-of-service attacks. Therefore, our protocol achieves the most important security, privacy and functional requirements of practical RFID systems for both adversaries with and without access to auxiliary information. We stress that our scheme only considers anonymity and untraceability of the communication between the tags and the reader that takes place when a tag is used to access some service. Therefore, our protocol does not consider the privacy of the communication between the tags and the anonymizers. Note that all tags access anonymizers and thus, an anonymization does not leak any information to the adversary about the use of a given tag accessing some service (i.e., when the tag communicates to a reader). Moreover, the use of services can be selective since only some tags can have access to some services and thus privacy is critical in this phase. Finally, note that the crucial issue is that the adversary must not be able to obtain any information about which tag accessed any service and about whether the same tag has obtained access to some services.

Our protocol provides basic availability, which means that the adversary cannot manipulate (i.e., invalidate) legitimate tags without physically attacking an anonymizer (and thus criminalizing himself). However, this is sufficient for most practical scenarios since a stolen or damaged public anonymizer can be detected and thus such attacks are unlikely to happen just to violate privacy. Further, public anonymizers can be physically secured (e.g., by a robust housing as it is used for surveillance cameras). Moreover, in the scenario of personal anonymizers, the damage that can be done by a corrupted anonymizer

is limited to violating the privacy of only the tags of one single user since only the key of this single user's anonymizer is revealed. A potential success in a security violation (i.e., in impersonating a legitimate tag) could motivate the adversary since he would obtain unauthorized access to services, which in turn means that he would get some economic advantages. However, our protocols are secure against impersonation attacks even against adversaries that corrupt anonymizers.

We do not consider unclonability of tags since this seems to be infeasible to achieve without hardware assumptions, which would significantly increase the costs of the tags. Further, we do not consider tracing or identification attacks based on the physical characteristics of tags, which in practice seem to be a general problem that cannot be prevented by protocols on the logical layer [48].

One of the main features of our scheme is its generic structure that allows to instantiate our scheme using various cryptographic primitives (i.e., any CPA-secure homomorphic encryption scheme) based on different number-theoretic assumptions with different performance properties. In particular, our protocol does not require tags to perform public-key cryptography (beyond the homomorphic operation that usually does not resort to modular exponentiations) and thus is not limited to the use of special lightweight public-key encryption schemes. This opens the possibility to employ optimized schemes, e.g., with short keys (in particular when using a prime modulus) and ciphertexts to reduce the memory requirements on tags⁵ and to decrease the size of the protocol messages.

5.3.1 Trust Model and Assumptions

Before presenting our anonymizer-enabled RFID system, we first give an informal description of the underlying trust relations that we formalize in Section 5.3.4. Following the majority of existing RFID models, we make the following assumptions.

Adversary \mathcal{A} . As in most RFID security models, we assume \mathcal{A} to control the wireless communication channel between readers, tags and anonymizers. This means that \mathcal{A} can eavesdrop, manipulate, delete and reroute all protocol messages sent by \mathcal{R} , \mathcal{T} and \mathcal{P} . Moreover, \mathcal{A} can obtain useful information (e.g., by visual observation) on whether \mathcal{R} accepted \mathcal{T} as a legitimate tag [99, 190].

⁵Typical low-end to mid-range RFID tags provide about 1–64 KBytes of memory.

Issuer \mathcal{I} . We assume \mathcal{I} to be trusted and that \mathcal{I} initializes tags and readers in a secure environment.

Readers \mathcal{R} . We assume that all readers are connected to the same backend system \mathcal{D} hosting a database DB. Thus, all honest readers have access to the same information and thus can be subsumed as *one single* reader entity \mathcal{R} . Moreover, \mathcal{R} can perform public-key cryptography and can handle multiple instances of the tag authentication protocol with different tags in parallel. As most RFID privacy models, we assume \mathcal{R} to be trusted. This means that \mathcal{R} behaves as intended and does nothing that violates the security and privacy goals of legitimate tags.

Tags \mathcal{T} . The tags considered are passive devices, which means that they do not have their own power supply and are powered by the electromagnetic field of the reader \mathcal{R} . Thus, tags cannot initiate communication, have a narrow communication range (e.g., a few centimeters to meters) and are constrained in their computational and storage capabilities, which limits them to basic cryptographic functions such as hashing, random number generation and symmetric-key encryption [9, 8, 141]. Tags are considered to be untrusted since the adversary \mathcal{A} can obtain full control of the tags and the data stored on them.

Anonymizers \mathcal{P} . Anonymizers can perform public-key cryptography and can handle multiple parallel instances of the anonymization protocol with different tags. Since a tag \mathcal{T} does not possess the required computational resources to update its state, it can always be tracked between two anonymizations. Therefore, to provide anonymity and unlinkability, it must be guaranteed that each tag is frequently anonymized by an honest anonymizer (e.g., every few minutes). In practice, this is achieved by a dense network of public anonymizers or a personal anonymizer. Observe that, in order to eavesdrop on every interaction of a tag with a reader or an anonymizer, the adversary \mathcal{A} must always be within the reading range of the tag. Due to the limited communication range of RFID this implies that \mathcal{A} is following the user of the tag, which obviously violates the tag user's privacy even if he would not carry an RFID tag. Thus, a privacy-preserving RFID system can at most offer privacy guarantees against adversaries that do not have permanent access to the tags. Moreover, an adversary in practice can at most corrupt a limited number of anonymizers, which ensures that there is at least one honest anonymizer

in the system. Hence, we consider anonymizers to be untrusted and assume that the adversary can get full control over many but not all anonymizers and their secrets.

5.3.2 Protocol Specification

Our RFID scheme consists of two protocols: the tag authentication and the tag anonymization protocol. The tag authentication protocol is executed by the reader \mathcal{R} and a tag \mathcal{T} and allows \mathcal{R} to check whether \mathcal{T} is legitimate. The tag anonymization protocol ensures anonymity and untraceability of \mathcal{T} in the authentication protocol by updating the authentication secrets of \mathcal{T} .

System Initialization

Reader setup. Given a security parameter $l_{\mathcal{R}} = (\mathfrak{h}, \mathfrak{n}) \in \mathbb{N}^2$, the reader \mathcal{R} generates a key pair $(sk_{\mathcal{R}}, pk_{\mathcal{R}}) \leftarrow \text{Genkey}^{\mathfrak{h}}(1^{\mathfrak{h}})$ for a CPA-secure homomorphic public-key encryption scheme (Definition 2.12). Moreover, \mathcal{R} initializes a secret database $\text{DB} \leftarrow \{\}$ that later stores the identities and authentication secrets of all legitimate tags. The secret key of \mathcal{R} is $sk_{\mathcal{R}}$ whereas the corresponding public key is $(\mathfrak{h}, \mathfrak{n}, pk_{\mathcal{R}})$. For brevity, we write $pk_{\mathcal{R}}$ to mean the complete tuple. Note that \mathfrak{n} denotes the bit length of the authentication secrets and nonces used in the tag authentication and the tag anonymization protocols.

Anonymizer setup. Given a security parameter $l_{\mathcal{P}} = (\mathfrak{a}, \mathfrak{n}) \in \mathbb{N}^2$, the anonymizer \mathcal{P} generates a key pair $(sk_{\mathcal{P}}, pk_{\mathcal{P}}) \leftarrow \text{Genkey}(1^{\mathfrak{a}})$ for the CPA-secure public-key encryption scheme (Definition 2.12). The secret key of \mathcal{P} is $sk_{\mathcal{P}}$ whereas the corresponding public key is the tuple $(\mathfrak{a}, \mathfrak{n}, pk_{\mathcal{P}})$. We write $pk_{\mathcal{P}}$ to mean the complete tuple.⁶

Tag setup. A tag \mathcal{T} with identifier ID is initialized by the issuer \mathcal{I} as follows: First, \mathcal{I} generates a long-term secret $K \xleftarrow{\$} \{0, 1\}^{\mathfrak{n}}$ and an ephemeral secret $T \xleftarrow{\$} \{0, 1\}^{\mathfrak{n}}$, which

⁶ As discussed in Section 5.1, there are two scenarios: public anonymizers and personal anonymizers. Since all public anonymizers have the same secret decryption key, they can be initialized with this key before they are deployed and the corresponding public key can be used with all tags in the system. Personal anonymizers (i.e., those running on the users' mobile phone or PDA) can have different user-specific keys. However, this requires the user of a personal anonymizer to register the public key of the personal anonymizer once with the tag issuing entity before he obtains the first tag that shall be anonymized with this anonymizer. For instance, in the application scenario of electronic transit tickets, the user may register the personal anonymizer online or at a ticket vending machine once before he purchases the very first ticket.

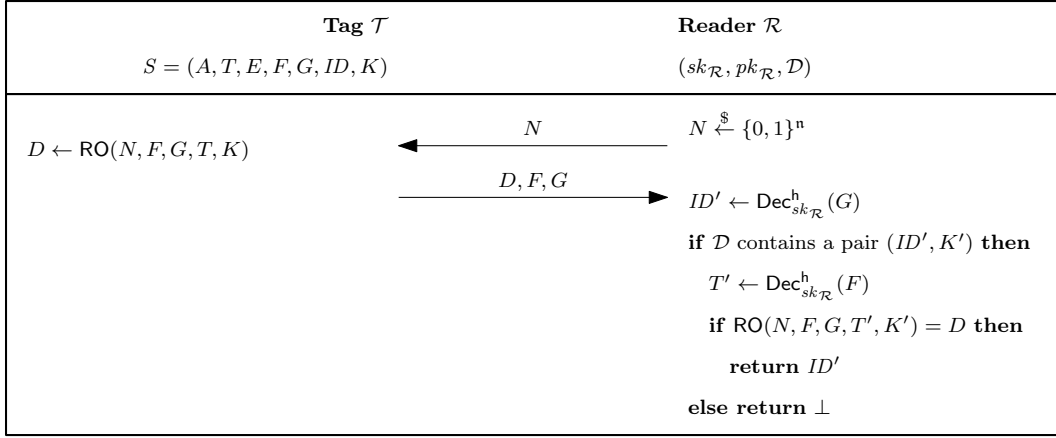


Figure 5.2: Privacy-Preserving Tag Authentication Protocol

are used later in the tag authentication protocol to authenticate \mathcal{T} to the reader \mathcal{R} . Moreover, \mathcal{I} generates a symmetric encryption key $A \leftarrow \text{Genkey}(1^s)$ for some $s \in \mathbb{N}$, which is used later by \mathcal{T} to encrypt the communication of the anonymization protocol. Further, \mathcal{I} encrypts $E \leftarrow \text{Enc}_{pk_{\mathcal{P}}}(A)$, $F \leftarrow \text{Enc}_{pk_{\mathcal{R}}}^h(T)$ and $G \leftarrow \text{Enc}_{pk_{\mathcal{R}}}^h(ID)$. The ciphertext E is used to transport the symmetric key A from \mathcal{T} to \mathcal{P} in the anonymization protocol whereas F and G are used to transport the ephemeral secret T and the tag identifier ID from \mathcal{T} to \mathcal{R} in the authentication protocol. Finally, \mathcal{I} updates the database $\text{DB} \leftarrow \text{DB} \cup \{(ID, K)\}$ of \mathcal{R} and initializes \mathcal{T} with the state $S \leftarrow (A, T, E, F, G, ID, K)$.

Tag Authentication Protocol

The authentication protocol (cf. Figure 5.2) is an interactive protocol between a tag \mathcal{T} with identifier ID and the reader \mathcal{R} with the goal to identify \mathcal{T} on the reader side. \mathcal{R} sends $N \xleftarrow{\$} \{0, 1\}^n$ to \mathcal{T} , which then computes $D \leftarrow \text{RO}(N, F, G, T, K)$, where RO is a random oracle (Definition 2.5) and responds with (D, F, G) . Then, \mathcal{R} decrypts $ID' \leftarrow \text{Dec}_{sk_{\mathcal{R}}}^h(G)$ and checks if its secret database DB contains a tuple (ID', K') . If this is the case, \mathcal{R} decrypts $T' \leftarrow \text{Dec}_{sk_{\mathcal{R}}}^h(F)$ and accepts \mathcal{T} by returning ID' only if $D = \text{RO}(N, F, G, T', K')$. Otherwise, \mathcal{R} rejects \mathcal{T} and returns \perp .

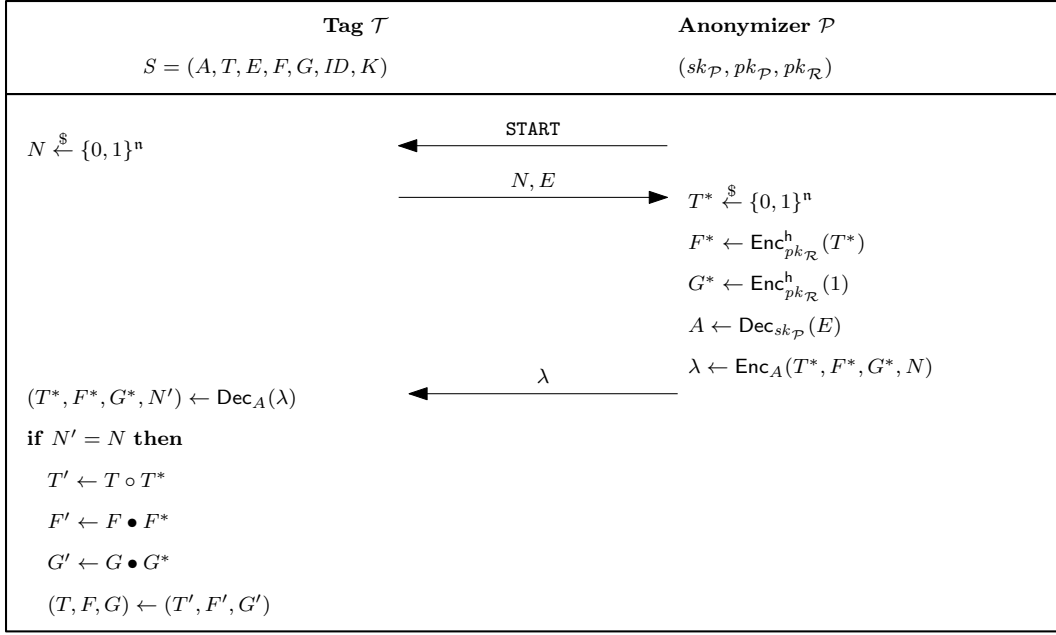


Figure 5.3: Tag Anonymization Protocol for Privacy-Preserving Authentication

Tag Anonymization Protocol

The tag anonymization protocol is illustrated in Figure 5.3. It is a protocol between a tag \mathcal{T} with identifier ID and an anonymizer \mathcal{P} with the goal to update the state S of \mathcal{T} . First, \mathcal{T} chooses $N \xleftarrow{\$} \{0, 1\}^n$ and sends (N, E) to \mathcal{P} . Then, \mathcal{P} chooses a new ephemeral tag secret T^* and encrypts it to $F^* \leftarrow \text{Enc}_{pk_{\mathcal{R}}}^h(T^*)$. Moreover, \mathcal{P} computes $G^* \leftarrow \text{Enc}_{pk_{\mathcal{R}}}^h(1)$ of the identity with regard to the homomorphic operation \circ of the public-key encryption scheme. Finally, \mathcal{P} decrypts $A \leftarrow \text{Dec}_{sk_{\mathcal{P}}}(E)$, computes $\lambda \leftarrow \text{Enc}_A(T^*, F^*, G^*, N)$ and sends λ to \mathcal{T} . Then, \mathcal{T} decrypts $(T^*, F^*, G^*, N') \leftarrow \text{Dec}_A(\lambda)$ and checks whether $N' = N$. If this is the case, \mathcal{T} computes a new ephemeral authentication secret $T' \leftarrow T \circ T^*$, the (homomorphic) public-key encryption $F' \leftarrow F \bullet F^*$ of the new ephemeral key T' and a new (re-randomized) encryption $G' \leftarrow G \bullet G^*$ of the tag identifier ID . Eventually, \mathcal{T} updates its state $(T, F, G) \leftarrow (T', F', G')$. If $N' \neq N$, \mathcal{T} aborts the anonymization protocol without updating its state.

5.3.3 Performance Evaluation

Using the (homomorphic) El Gamal public-key encryption scheme [60], our protocol requires tags to provide about 0.6 KBytes of non-volatile memory. Anonymization requires the tag to generate a random number, decrypt one symmetric ciphertext and to perform five modular multiplications. Tag authentication requires the tag to evaluate a hash function. Note that the anonymization protocol is completely transparent to the user whereas identification usually requires the user to wait (e.g., at a door) until the authentication protocol completes. Thus, in contrast to the anonymization protocol, most practical applications have strict time constraints on the identification protocol. Our scheme should be implementable with widely available RFID tags.

5.3.4 Security Analysis

To formalize and to prove the security and privacy properties of our scheme, an appropriate security and privacy model is needed. Since existing RFID security and privacy models do not capture anonymizer-enabled protocols (cf. Section 3.3.3), we extend the model by Vaudenay [190] (V-Model) to the first universal security and privacy model for anonymizer-enabled RFID systems with trusted readers.

The V-Model is very similar to the PV-Model (cf. Section 4.2) with the only difference that the V-Model does not include reader authentication. Specifically the differences of the V-Model to the PV-Model are that (1) in the **AuthTag** protocol (cf. Definition 4.1), the tag \mathcal{T} does not produce any output and (2) there is no definition of reader authentication, i.e., Definition 4.5 does not exist in the V-Model.

System Model

To form the anonymizer-enabled model, the V-Model must be extended to consider the anonymizer \mathcal{P} and the corresponding protocols. This means that there must be a procedure to set up \mathcal{P} and an interactive protocol, where \mathcal{P} updates the state of the tags. Following the V-Model [190], we define an anonymizer-enabled RFID system as follows:

Definition 5.1 (Anonymizer-Enabled RFID System). *An anonymizer-enabled RFID system $(\mathcal{R}, \mathcal{T}, \mathcal{P}, \text{SetupReader}, \text{SetupAnon}, \text{SetupTag}, \text{AnonTag}, \text{AuthTag})$ is a tuple of p.p.t. algorithms that are defined as follows:*

$\text{SetupReader}(1^{l_{\mathcal{R}}}) \rightarrow (sk_{\mathcal{R}}, pk_{\mathcal{R}}, \text{DB})$ On input of a security parameter $l_{\mathcal{R}}$, this function initializes the reader \mathcal{R} by creating some public parameters $pk_{\mathcal{R}}$ that are known to all entities and some secret parameters $sk_{\mathcal{R}}$ that are only known to \mathcal{R} . This function also creates a secret database DB that can only be accessed by \mathcal{R} and that stores the identities and authentication secrets of all legitimate tags.

$\text{SetupAnon}(1^{l_{\mathcal{P}}}, pk_{\mathcal{R}}) \rightarrow (sk_{\mathcal{P}}, pk_{\mathcal{P}})$ On input of a security parameter $l_{\mathcal{P}}$ and the public key $pk_{\mathcal{R}}$ of \mathcal{R} , this function initializes the anonymizer \mathcal{P} by creating some public parameters $pk_{\mathcal{P}}$ that are known to all entities and some secret parameters $sk_{\mathcal{P}}$ that are only known to \mathcal{P} .

$\text{SetupTag}_{pk_{\mathcal{R}}}(ID, pk_{\mathcal{P}}) \rightarrow (K, S)$ This function generates a tag-specific secret K and uses the public key $pk_{\mathcal{R}}$ of \mathcal{R} to create an initial state S for the tag \mathcal{T} with identifier ID . \mathcal{T} is initialized with S and (ID, K) is stored in the database DB of \mathcal{R} . Since \mathcal{T} must authenticate the anonymizer \mathcal{P} in the anonymization protocol, this procedure involves $pk_{\mathcal{P}}$.

$\text{AnonTag}[\mathcal{T} : S; \mathcal{P} : sk_{\mathcal{P}}; * : pk_{\mathcal{P}}, pk_{\mathcal{R}}] \rightarrow [\mathcal{T} : S'; \mathcal{P} : S']$ This is an interactive protocol that is (frequently) run between the tag \mathcal{T} with identifier ID and the anonymizer \mathcal{P} to update the state S of \mathcal{T} to a new indistinguishable state S' .

$\text{AuthTag}[\mathcal{T} : S; \mathcal{R} : sk_{\mathcal{R}}, \text{DB}; * : pk_{\mathcal{R}}] \rightarrow [\mathcal{T} : -; \mathcal{R} : \text{out}]$ This is an interactive protocol between the tag \mathcal{T} with identifier ID and the reader \mathcal{R} . The goal of this protocol is to identify \mathcal{T} and to verify whether \mathcal{T} is legitimate. With overwhelming probability, \mathcal{R} returns $\text{out} = ID$ if \mathcal{T} is legitimate and $\text{out} = \perp$ otherwise.⁷

Definition 5.2 (Correctness). An RFID system (Definition 5.1) achieves correctness if $\forall l, \forall (sk_{\mathcal{R}}, pk_{\mathcal{R}}, \text{DB}) \in [\text{SetupReader}(1^l)], \forall (sk_{\mathcal{P}}, pk_{\mathcal{P}}) \in [\text{SetupAnon}(1^{l_{\mathcal{P}}}, pk_{\mathcal{R}})], \forall (K, S) \in [\text{SetupTag}_{pk_{\mathcal{R}}}(ID, pk_{\mathcal{P}})]$ and $\forall S'$ where $\text{AnonTag}[\mathcal{T} : S; \mathcal{P} : sk_{\mathcal{P}}; * : pk_{\mathcal{P}}, pk_{\mathcal{R}}] \rightarrow [\mathcal{T} : S'; \mathcal{P} : S']$, it holds that $\text{Auth}[\mathcal{T} : S'; \mathcal{R} : sk_{\mathcal{R}}, \text{DB}; * : pk_{\mathcal{R}}] \rightarrow [\mathcal{T} : -; \mathcal{R} : ID]$ with overwhelming probability.

⁷ A false negative occurs when \mathcal{T} is legitimate but $\text{out} = \perp$. A false positive happens if \mathcal{T} is not legitimate and $\text{out} \neq \perp$. An incorrect identification occurs if the tag \mathcal{T} with identifier ID is legitimate but $\text{out} \notin \{ID, \perp\}$

Adversary Model

The V-Model [190] defines the privacy and security objectives as a security experiment, where a polynomially bounded adversary \mathcal{A} can interact with a set of oracles that model the capabilities of \mathcal{A} . In the anonymizer-enabled model, \mathcal{A} may obtain information from the anonymization protocol. This ability is modeled by allowing \mathcal{A} to launch new anonymization protocol sessions and to interact with the anonymizer. To consider the case where \mathcal{A} controls a set of anonymizers, we allow \mathcal{A} to obtain the secrets of the anonymizers by corrupting them. In the anonymizer-enabled model, \mathcal{A} has access to the following oracles:

CreateTag^b(ID, pk_P) This oracle allows the adversary \mathcal{A} to set up a tag \mathcal{T} with identifier ID . This oracle internally calls **SetupTag_{pk_R}(ID, pk_P)** to create (K, S) for \mathcal{T} . If input $b = 1$, \mathcal{A} chooses \mathcal{T} to be legitimate, which means that (ID, K) is added to the database DB of the reader \mathcal{R} . For input $b = 0$, \mathcal{A} can create illegitimate tags where (ID, K) is *not* added to DB. This models the fact that the adversary can obtain (e.g., buy) legitimate tags and create forgeries.

DrawTag(δ) \rightarrow (vtag₁, b₁, ..., vtag_n, b_n) Initially, the adversary \mathcal{A} cannot interact with any tag but must query the **DrawTag** oracle to get access to a set of tags that has been chosen according to a given tag distribution δ . This models the fact that \mathcal{A} can only interact with the tags within his reading range. \mathcal{A} usually knows the tags he can interact with only by some temporary tag identifiers vtag₁, ..., vtag_n (e.g., in our protocol the tuple (F, G) can be seen as temporary tag identifier). The **DrawTag** oracle manages a secret look-up table Γ that keeps track of the real identifier ID_i that is associated with each temporary tag identifier vtag_i, i.e., $\Gamma(\text{vtag}_i) = ID_i$. Moreover, the **DrawTag** oracle also provides \mathcal{A} with information on whether the corresponding tags are legitimate ($b_i = 1$) or not ($b_i = 0$). This models the availability of auxiliary information to \mathcal{A} .⁸

FreeTag(vtag) Contrary to the **DrawTag** oracle, the **FreeTag** oracle makes a tag vtag inaccessible to the adversary \mathcal{A} , which means that \mathcal{A} cannot interact with vtag any longer until it is made accessible again (under a new temporary tag identifier

⁸ For instance, in an access control scenario, the adversary may notice that a tag vtag_i is legitimate by observing its communication with a reader at a locked door and then watching whether the door opens or not.

$vtag'$) by another **DrawTag** query. This models the fact that a tag can get out of the reading range of the adversary.

LaunchIdent $(\cdot) \rightarrow \pi_{\mathcal{R}}$ This oracle makes the reader \mathcal{R} to start a new instance $\pi_{\mathcal{R}}$ of the **AuthTag** protocol, which allows the adversary \mathcal{A} to start different parallel **AuthTag** protocol instances with \mathcal{R} .

LaunchAnon $(\cdot) \rightarrow \pi_{\mathcal{P}}$ This oracle makes the anonymizer \mathcal{P} to start a new instance $\pi_{\mathcal{P}}$ of the **AnonTag** protocol, which allows the adversary \mathcal{A} to start different parallel **AnonTag** protocol instances with \mathcal{P} .

SendTag $(m, vtag) \rightarrow m'$ This oracle sends a message m to the tag \mathcal{T} that is known as $vtag$ to the adversary \mathcal{A} . The tag \mathcal{T} responds with a message m' . This allows \mathcal{A} to perform active attacks against both the **AnonTag** and the **AuthTag** protocol.

SendReader $(m, \pi_{\mathcal{R}}) \rightarrow m'$ This oracle sends a message m to the instance $\pi_{\mathcal{R}}$ of the **AuthTag** protocol that is executed by the reader \mathcal{R} , which responds with a message m' . This allows \mathcal{A} to perform active attacks against the **AuthTag** protocol.

SendAnon $(m, \pi_{\mathcal{P}}) \rightarrow m'$ This oracle sends a message m to the instance $\pi_{\mathcal{P}}$ of the **AnonTag** protocol that is executed by an honest anonymizer \mathcal{P} , which responds with a message m' . This allows \mathcal{A} to perform active attacks against the **AnonTag** protocol.

Result $(\pi_{\mathcal{R}})$ This oracle returns 1 if the instance $\pi_{\mathcal{R}}$ of the **AuthTag** protocol has been completed but the tag \mathcal{T} that participates in the protocol has not been accepted by the reader \mathcal{R} . In case \mathcal{R} identified a legitimate tag, **Result** returns 0. This allows the adversary \mathcal{A} to obtain auxiliary information on whether the authentication of \mathcal{T} was successful or not.

CorruptTag $(vtag) \rightarrow S$ This oracle returns the current state S of the tag \mathcal{T} that is known as $vtag$ to the adversary \mathcal{A} . This models (physical) attacks on tags that disclose the current tag state.

CorruptAnon $(\mathcal{P}) \rightarrow (sk_{\mathcal{P}})$ This oracle returns the secret parameter $sk_{\mathcal{P}}$ of the anonymizer \mathcal{P} . This models (physical) attacks against honest anonymizers that disclose the secret $sk_{\mathcal{P}}$ of \mathcal{P} .

Assumptions. As discussed in Section 5.3.1, we assume that there is at least one honest anonymizer in the system whose communication cannot be eavesdropped or manipulated by the adversary.

Assumption 5.1 (Honest Anonymization). *A tag \mathcal{T} with identifier ID always runs **AnonTag** with an honest anonymizer \mathcal{P} at least once before each execution of **AuthTag** with the reader \mathcal{R} and before each **CorruptTag**($vtag$) query where $\Gamma(vtag) = ID$.*

Adversary classes. Following the V-Model [190], we distinguish different adversary classes that represent adversaries of different strength.

Definition 5.3 (Adversary Classes). *An adversary is a p.p.t. algorithm that has arbitrary access to all oracles described above. Weak adversaries cannot access the **CorruptTag** oracle. Forward adversaries cannot query any other oracle than **CorruptTag** after they made the first **CorruptTag** query. Destructive adversaries cannot query any oracle for $vtag$ again after they made a **CorruptTag**($vtag$) query. Strong adversaries have no restrictions on the use of the **CorruptTag** oracle. Narrow adversaries cannot access the **Result** oracle.*

Note that Definition 5.3 is very similar to Definition 4.3 but additionally allows the adversary to interact with the **LaunchAnon**, **SendAnon** and **CorruptAnon** oracles required for considering the anonymizer.

Tag Authentication

The definition of tag authentication of the V-Model is very similar to Definition 4.4 and considers attacks where the adversary aims to impersonate or to forge a legitimate tag. More precisely, the definition is based on a security experiment $\mathbf{Exp}_{\mathcal{A}_{\text{sec}}}^{\text{sec}}$ where a *strong adversary* \mathcal{A}_{sec} must create an instance $\pi_{\mathcal{R}}$ of the **AuthTag** protocol with the reader \mathcal{R} and finish this protocol instance with a query **SendReader**($m, \pi_{\mathcal{R}}$). Note that \mathcal{A}_{sec} can arbitrarily interact with all of the oracles underlying Definition 5.3 at any time during the experiment. \mathcal{A}_{sec} wins if (1) \mathcal{R} identifies a legitimate tag ID in the instance $\pi_{\mathcal{R}}$ of the **AuthTag** protocol, (2) tag ID has not been corrupted and (3) tag ID and \mathcal{R} have not run any instance $\pi_{\mathcal{R}}'$ of the **AuthTag** protocol that generated the same messages as in instance $\pi_{\mathcal{R}}$ (i.e., $\pi_{\mathcal{R}}$ is not a *replay* of an old transcript $\pi_{\mathcal{R}}'$). Let $\mathbf{Exp}_{\mathcal{A}_{\text{sec}}}^{\text{sec}} = 1$ denote the case where \mathcal{A}_{sec} wins the security experiment.

Definition 5.4 (Tag Authentication [190]). *An RFID system (Definition 5.1) achieves tag authentication if for any strong adversary \mathcal{A}_{sec} (Definition 5.3) $\Pr[\mathbf{Exp}_{\mathcal{A}_{\text{sec}}}^{\text{sec}} = 1]$ is negligible.*

Definition 5.4 can be used in the anonymizer-enabled model with the addition that, during the security experiment $\mathbf{Exp}_{\mathcal{A}_{\text{sec}}}^{\text{sec}}$, \mathcal{A}_{sec} is allowed to interact with all the oracles in Section 5.3.4. In particular, \mathcal{A}_{sec} can corrupt all the anonymizers which models the fact that anonymizers should not be able to clone or to forge tags.

Privacy Definition

The privacy definition of the V-Model is similar to Definition 4.6. It is very flexible and, dependent on the class of adversaries considered (Definition 5.3), it covers different notions of privacy. For strong adversaries the definition considers anonymity, backward and forward untraceability. Therefore, the privacy definition requires the communication of a tag \mathcal{T} to not reveal any information that helps the adversary \mathcal{A}_{prv} to trace or to identify \mathcal{T} . The privacy definition is based on the existence of a simulator \mathcal{B} that can simulate the communication of \mathcal{T} to \mathcal{A}_{prv} without using any of the secrets of the RFID system. \mathcal{B} must answer all queries of \mathcal{A}_{prv} by only using the inputs and outputs of the oracle queries that \mathcal{A}_{prv} previously made (i.e., \mathcal{B} “sees” what \mathcal{A}_{prv} “sees”). In case the success probability of \mathcal{A}_{prv} does not change significantly when interacting with \mathcal{B} instead of the real RFID system, the communication of \mathcal{T} does not help \mathcal{A}_{prv} to break the privacy properties of the RFID scheme. \mathcal{B} is called *blinder* [190] and $\mathcal{A}_{\text{prv}}^{\mathcal{B}}$ who interacts with \mathcal{B} is called *blinded adversary*.

More formally, the privacy definition considers a security experiment $\mathbf{Exp}_{\mathcal{A}_{\text{prv}}}^{\text{prv}}$ where the adversary \mathcal{A}_{prv} must distinguish whether he interacts with the real RFID system or the blinder \mathcal{B} . Therefore, \mathcal{A}_{prv} first performs an attack phase that is followed by an analysis phase. In the attack phase, \mathcal{A}_{prv} is allowed to interact with the oracles underlying Definition 5.3 in an arbitrary way. In the analysis phase, \mathcal{A}_{prv} cannot access the oracles any more but is given access to the secret table Γ of the **DrawTag** oracle, which allows \mathcal{A}_{prv} to link the temporary identifiers *vtag* of all the tags he interacted with to their corresponding real identities *ID*. Finally, \mathcal{A}_{prv} must return a bit b to indicate whether he interacted with the blinder \mathcal{B} ($b = 1$) or the real RFID system ($b = 0$). This leads to the following privacy definition:

Definition 5.5 (Privacy [190]). *Let C be one of the adversary classes of Definition 5.3. An RFID system (Definition 5.1) is C -private if for any adversary \mathcal{A}_{prv} of class C there exists a blinder \mathcal{B} such that $|\Pr[\mathbf{Exp}_{\mathcal{A}_{\text{prv}}}^{\text{prv}} = 1] - \Pr[\mathbf{Exp}_{\mathcal{A}_{\text{prv}}}^{\mathcal{B}} = 1]|$ is negligible.*

The communication of a tag with the reader is modeled by the **LaunchIdent**, **SendReader**, **SendTag** and the **Result** oracle. Thus, the blinder \mathcal{B} must simulate these oracles. In the anonymizer-enabled model, we additionally have the **LaunchAnon** and the **SendAnon** oracles that model the interaction of a tag with the anonymizer. However, we are not concerned of the privacy of the communication between the tags and the anonymizer. Thus, \mathcal{A}_{prv} has no access to the **LaunchAnon** and the **SendAnon** oracle, which hence need not to be simulated by \mathcal{B} . Note that the **CorruptTag** query is not simulated by \mathcal{B} because Definition 5.5 only captures the privacy loss of the wireless communication of tags.

Security Proof

We are now ready to formally state the security and privacy properties of the protocol presented in Section 5.3.2 in the following theorem.

Theorem 5.1 (Correctness, Security and Privacy). *The RFID system presented in Section 5.3.2 is correct (Definition 5.2), provides tag authentication (Definition 5.4) in the random oracle model (Definition 2.5) and it is narrow-strong and forward private (Definition 5.5) in the random oracle model under Assumption 5.1 if the homomorphic public-key encryption scheme is CPA-secure (Definition 2.12).*

Note that Assumption 5.1 is *only* required to ensure the privacy properties of our scheme. Security against impersonation attacks also holds if there is no (honest) anonymizer in the system.

Proof of correctness (Theorem 5.1). No false negative can be produced since each legitimate tag \mathcal{T} will always be accepted by the reader \mathcal{R} . A false positive cannot be produced since the decryption of G outputs a unique ID and, if ID is not in the database DB , \mathcal{R} immediately rejects the identification. \square

Proof of tag authentication (Theorem 5.1). The idea of the security proof is as follows: By contradiction, we assume that there is a narrow-strong adversary \mathcal{A}_{sec} (Definition 5.3), who wins the security game of Definition 5.4. Given \mathcal{A}_{sec} , one can construct a p.p.t. algorithm that finds a collision to the random oracle with non-negligible probability.

However, by the pseudo-randomness of the random oracle, this can happen with at most negligible probability.

More detailed, assume by contradiction that \mathcal{A}_{sec} succeeds in the security experiment $\text{Exp}_{\mathcal{A}_{\text{sec}}}^{\text{sec}}$. This means that, at some point during the experiment, there is a transcript $\pi_{\mathcal{R}} = (N, (D, F, G))$ of the **AuthTag** protocol that has been generated by \mathcal{A}_{sec} , where the reader \mathcal{R} decrypts F and G with its secret key $sk_{\mathcal{R}}$ and gets an ephemeral tag secret T and a tag identifier ID for which there is a tuple (ID, K) in the database DB of \mathcal{R} such that $\text{RO}(N, F, G, T, K) = D$. Moreover, $\pi_{\mathcal{R}}$ is different from all other **AuthTag** protocol transcripts that have been generated by \mathcal{A}_{sec} during the experiment and the tag associated with ID has not been corrupted. Since $\pi_{\mathcal{R}} = (N, (D, F, G))$ is a new transcript of the **AuthTag** protocol we distinguish the following two cases:

CASE 1: \mathcal{A}_{sec} replays D from a previously recorded protocol transcript. In the first case, D is the output of a previous **SendTag** query. Let $(N', (D, F', G'))$ be the transcript generated by the tag that answered to such a query. Clearly, we have that $D = \text{RO}(N', F', G', T', K')$ for some T' and K' . Moreover we also have that $D = \text{RO}(N, F, G, T, K)$ since \mathcal{A}_{sec} is successful and this is checked by \mathcal{R} . However the assumption that $\pi_{\mathcal{R}} = (N, (D, F, G))$ is a new transcript, implies that either $N' \neq N$ or $F' \neq F$ or $G' \neq G$ and Therefore, \mathcal{A}_{sec} and the tag together can compute two different inputs to the random oracle that result in the same output. According to Definition 2.5, this clearly can happen only with probability $\leq 2^{-n/2}$, which is negligible.

CASE 2: \mathcal{A}_{sec} forges D . In the second case, D has never been the output of a previous **SendTag** query. Observe that the output of the random oracle is a random string. In this case, since RO is a random oracle and the reader verifies that $\text{RO}(N, F, G, T, K) = D$, we have that D can be correctly computed only by guessing it (which however occurs with negligible probability of 2^{-n}) or through a random oracle query. However, the latter requires the correct guess of K that previously has never been used for the computation of any message with the only exception of previous queries to the random oracle. Since the outputs of the random oracle are independent of their inputs, the output of the random oracle leaks no information on K . Thus, the probability of guessing the correct value of K is $\leq 2^{-n}$, which is negligible. \square

Proof of narrow-strong privacy (Theorem 5.1). The idea of the privacy proof is as follows: By contradiction, we assume that there is a narrow-strong adversary \mathcal{A}_{prv} (Definition 5.3), who wins the experiment of Definition 5.5 with non-negligible probability.

Given such an adversary \mathcal{A}_{prv} , one can construct a p.p.t. algorithm that breaks the CPA-security (Definition 2.12) of the homomorphic public key encryption scheme with non-negligible probability. However, since the encryption scheme is assumed to be CPA-secure, this can happen with at most negligible probability, which is a contradiction. More detailed, we prove by contradiction that for any narrow-strong adversary \mathcal{A}_{prv} there exists a blinder \mathcal{B} such that \mathcal{A}_{prv} has no significant advantage over the blinded adversary $\mathcal{A}_{\text{prv}}^{\mathcal{B}}$.

The construction of \mathcal{B} is as follows. Clearly, all **LaunchIdent** and **SendReader** oracles are trivial to simulate since no **Result** oracle query is allowed. Further, all queries of the form **SendTag**($\lambda, vtag$) are forwarded to the real oracles. However, queries of the form **SendTag**($N, vtag$), are part of the **AuthTag** protocol and thus, their responses (D, F, G) must be simulated by \mathcal{B} . To simulate the ciphertext G , \mathcal{B} chooses a random tag identifier ID and computes $G \leftarrow \text{Enc}_{pk_{\mathcal{R}}}^h(ID)$. \mathcal{B} simulates F by encrypting a random secret T to $F \leftarrow \text{Enc}_{pk_{\mathcal{R}}}^h(T)$. The value D is simulated by a randomly chosen value from the output domain of **RO**. In the following, we show that the simulation of the ephemeral secrets T and their corresponding ciphertexts F as well as of the ciphertexts G is perfect due to the **RO** and the CPA-security of the homomorphic public-key encryption scheme, respectively.

Let $\mathbf{B}(\mathcal{A}_{\text{prv}}^{\mathcal{B}})$ denote the event that $\mathcal{A}_{\text{prv}}^{\mathcal{B}}$ detects the presence of \mathcal{B} after a polynomial number of j oracle queries. Moreover, let $\mathbf{M}(\mathcal{A}_{\text{prv}}^{\mathcal{B}})$ be the event that at least one of the queries that has been made by $\mathcal{A}_{\text{prv}}^{\mathcal{B}}$ to the oracle **RO** matches one of the queries that has been made by the **SendTag** oracle to respond to some **SendTag**($N, vtag$) query (resp. that should have been made by the **SendTag** oracle if it would not have been blinded). Note that the simulation by \mathcal{B} is perfect when $|\Pr[\mathbf{B}(\mathcal{A}_{\text{prv}}^{\mathcal{B}})] - \Pr[\mathbf{B}(\mathcal{A}_{\text{prv}})]| = 0$. Observe that

$$\Pr[\mathbf{B}(\mathcal{A}_{\text{prv}}^{\mathcal{B}})] = \Pr[\mathbf{B}(\mathcal{A}_{\text{prv}}^{\mathcal{B}}) | \mathbf{M}(\mathcal{A}_{\text{prv}}^{\mathcal{B}})] \cdot \Pr[\mathbf{M}(\mathcal{A}_{\text{prv}}^{\mathcal{B}})] + \Pr[\mathbf{B}(\mathcal{A}_{\text{prv}}^{\mathcal{B}}) | \neg \mathbf{M}(\mathcal{A}_{\text{prv}}^{\mathcal{B}})] \cdot \Pr[\neg \mathbf{M}(\mathcal{A}_{\text{prv}}^{\mathcal{B}})].$$

Note that $\Pr[\mathbf{B}(\mathcal{A}_{\text{prv}}^{\mathcal{B}}) | \mathbf{M}(\mathcal{A}_{\text{prv}}^{\mathcal{B}})] \leq 1$ since in this case $\mathcal{A}_{\text{prv}}^{\mathcal{B}}$ can distinguish the output of the real **SendTag** oracle from the output of \mathcal{B} . Furthermore, we can estimate $\Pr[\mathbf{M}(\mathcal{A}_{\text{prv}}^{\mathcal{B}})]$ as follows: Note that in this case $\mathcal{A}_{\text{prv}}^{\mathcal{B}}$ can distinguish the output (D, F, G) of the real oracle from the output of \mathcal{B} . Since $\mathcal{A}_{\text{prv}}^{\mathcal{B}}$ interacts with \mathcal{B} , D is a random value whereas in the case of the real oracles $D = \text{RO}(N, F, G, T, K)$. Observe that $\mathcal{A}_{\text{prv}}^{\mathcal{B}}$ knows N , F and G since they have been sent by the **SendReader** and **SendTag** oracles, respectively. Moreover, since $\mathcal{A}_{\text{prv}}^{\mathcal{B}}$ is a strong adversary, he knows K from the **CorruptTag** oracle.

However, $\mathcal{A}_{\text{prv}}^{\mathcal{B}}$ does not know T since, due to Assumption 5.1, the state (T, F, G) of each tag will be updated to an indistinguishable state (T', F', G') before each execution of the **AuthTag** protocol and before the tag can be corrupted by $\mathcal{A}_{\text{prv}}^{\mathcal{B}}$. Thus, if $\mathcal{A}_{\text{prv}}^{\mathcal{B}}$ created n legitimate tags and made q queries to the oracle **RO**, then $\Pr [\mathbf{M}(\mathcal{A}_{\text{prv}}^{\mathcal{B}})] \leq nqj2^{-n}$, which is negligible. It follows that

$$\begin{aligned} \Pr [\mathbf{B}(\mathcal{A}_{\text{prv}}^{\mathcal{B}})] &\leq nqj2^{-n} + (1 - nqj2^{-n}) \cdot \Pr [\mathbf{B}(\mathcal{A}_{\text{prv}}^{\mathcal{B}}) | \neg \mathbf{M}(\mathcal{A}_{\text{prv}}^{\mathcal{B}})] \\ &\leq \Pr [\mathbf{B}(\mathcal{A}_{\text{prv}}^{\mathcal{B}}) | \neg \mathbf{M}(\mathcal{A}_{\text{prv}}^{\mathcal{B}})]. \end{aligned}$$

Similar arguments lead to $\Pr [\mathbf{B}(\mathcal{A}_{\text{prv}})] \leq \Pr [\mathbf{B}(\mathcal{A}_{\text{prv}}) | \neg \mathbf{M}(\mathcal{A}_{\text{prv}})]$ and it follows that

$$|\Pr [\mathbf{B}(\mathcal{A}_{\text{prv}}^{\mathcal{B}})] - \Pr [\mathbf{B}(\mathcal{A}_{\text{prv}})]| \leq |\Pr [\mathbf{B}(\mathcal{A}_{\text{prv}}^{\mathcal{B}}) | \neg \mathbf{M}(\mathcal{A}_{\text{prv}}^{\mathcal{B}})] - \Pr [\mathbf{B}(\mathcal{A}_{\text{prv}}) | \neg \mathbf{M}(\mathcal{A}_{\text{prv}})]|.$$

This means that $\mathcal{A}_{\text{prv}}^{\mathcal{B}}$ detects \mathcal{B} based on value D only with negligible probability. Moreover, due to Assumption 5.1, the simulation of the ephemeral secrets T and their corresponding ciphertexts F is perfect. In the following, we show that if the homomorphic public-key encryption scheme is CPA-secure, \mathcal{A}_{sec} detects the simulation of the ciphertexts G with at most negligible probability.

Let $\mathbf{BM}(\mathcal{A}_{\text{prv}})$ denote the event $\mathbf{B}(\mathcal{A}_{\text{prv}}) | \neg \mathbf{M}(\mathcal{A}_{\text{prv}})$. Further, assume that there is a strong adversary \mathcal{A}_{prv} such that $\Pr [\mathbf{BM}(\mathcal{A}_{\text{prv}})]$ is negligible whereas $\Pr [\mathbf{BM}(\mathcal{A}_{\text{prv}}^{\mathcal{B}})]$ is non-negligible. This means that we assume

$$|\Pr [\mathbf{BM}(\mathcal{A}_{\text{prv}})] - \Pr [\mathbf{BM}(\mathcal{A}_{\text{prv}}^{\mathcal{B}})]| \quad (5.1)$$

to be non-negligible, which contradicts narrow-strong privacy (Definition 5.5).

Now consider the following partial blinder \mathcal{B}_i where the first i oracle queries of the form **SendTag**($N, vtag$) are answered by \mathcal{B} and all other oracle queries are answered by the real oracles. Note that \mathcal{B}_0 forwards all oracle queries to the real oracles whereas \mathcal{B}_j simulates all oracle queries. It follows that $|\Pr [\mathbf{BM}(\mathcal{A}_{\text{prv}}^{\mathcal{B}_0})] - \Pr [\mathbf{BM}(\mathcal{A}_{\text{prv}}^{\mathcal{B}_j})]|$ is non-negligible. Using hybrid arguments it follows that there must be an index i such that

$$|\Pr [\mathbf{BM}(\mathcal{A}_{\text{prv}}^{\mathcal{B}_{i-1}})] - \Pr [\mathbf{BM}(\mathcal{A}_{\text{prv}}^{\mathcal{B}_i})]| \quad (5.2)$$

is non-negligible. Given this index i , we can use \mathcal{A}_{prv} to construct the following CPA-distinguisher \mathcal{C}^{cpa} (cf. Definition 2.12) for the homomorphic public-key encryption scheme

as follows: \mathcal{C}^{cpa} simulates the RFID scheme to \mathcal{A}_{prv} with the following deviations. First, the key pair $(sk_{\mathcal{R}}, pk_{\mathcal{R}})$ of the reader \mathcal{R} is chosen by the CPA-challenger of the homomorphic public-key encryption scheme. Second, \mathcal{C}^{cpa} works as the blinder \mathcal{B}_i (or the blinder \mathcal{B}_{i-1}). This means that \mathcal{C}^{cpa} simulates the first $i - 1$ responses to the $\text{SendTag}(N, vtag)$ oracle as the blinder \mathcal{B}_{i-1} does. Moreover, \mathcal{C}^{cpa} simulates the response (D_i, F_i, G_i) to the i -th oracle query $\text{SendTag}(N_i, vtag_i)$ as follows. First, \mathcal{C}^{cpa} queries the CPA-challenger with two messages $m_0 = ID$ and $m_1 = ID'$ where ID is the real identifier of the tag \mathcal{T} that is associated with $vtag_i$ and ID' is a random tag identifier. The CPA-challenger then responds with a ciphertext G_i that encrypts either m_0 or m_1 under the public key $pk_{\mathcal{R}}$ of the reader \mathcal{R} . Note that \mathcal{C}^{cpa} knows the long-term secret K , the ephemeral secret T_i and the ciphertext F_i of the tag associated with $vtag_i$ since \mathcal{C}^{cpa} simulates the DrawTag and CreateTag oracles and the oracles of the AnonTag protocol. This allows \mathcal{C}^{cpa} to compute $D \leftarrow \text{RO}(N_i, F_i, G_i, T_i, K)$. All other $\text{SendTag}(N, vtag)$ oracle queries are forwarded to the real SendTag oracle.

Note that if G_i encrypts m_0 , \mathcal{C}^{cpa} simulates \mathcal{B}_{i-1} whereas in case G_i encrypts m_1 , \mathcal{C}^{cpa} simulates \mathcal{B}_i (note that F_i and D_i are correctly simulated with overwhelming probability and thus, \mathcal{A}_{prv} cannot distinguish whether it interacts with the blinder or the real oracle). Due to Equation 5.2, we know that \mathcal{A}_{prv} detects \mathcal{B}_i with non-negligible probability whereas \mathcal{A}_{prv} detects \mathcal{B}_{i-1} with at most negligible probability. Thus, when \mathcal{A}_{prv} returns $b = 1$, \mathcal{C}^{cpa} must have simulated \mathcal{B}_i , which means that G_i encrypts m_0 . Clearly, this violates the CPA-security of the homomorphic public-key encryption scheme, which therefore guarantees that Equation 5.2 is negligible. Moreover, since \mathcal{A}_{prv} can at most make a polynomial number of j queries, this implies that Equation 5.1 must be negligible.

Since we have shown that \mathcal{A}_{prv} detects the simulation of the tuples (D, F, G) with at most negligible probability and all other oracles are simulated perfectly, \mathcal{A}_{prv} detects the blinder \mathcal{B} with at most negligible probability. \square

Proof of forward privacy (Theorem 5.1). To prove forward-privacy, we can use the following lemma:

Lemma 5.1 (Narrow-Forward Privacy Implies Forward-Privacy [190]). *For every secure RFID scheme that has the property that, whenever a legitimate tag \mathcal{T} and the reader \mathcal{R} have executed a complete run of the AuthTag protocol in a secure environment (i.e., where no adversary can manipulate the protocol-run), the output out of \mathcal{R} is never \perp*

(i.e., \mathcal{R} does never reject a legitimate tag), it holds that narrow-forward privacy implies forward-privacy.

According to Theorem 5.1, our scheme is narrow-strong private, which implies narrow-forward privacy. Further, it is correct and secure, which means that it fulfills all requirements to apply Lemma 5.1. Since the original proof of Lemma 5.1 is also valid in the anonymizer-enabled model, we can apply Lemma 5.1 to prove that our scheme achieves forward privacy. \square

5.4 Anonymizers in RFID Systems with Untrusted Readers

In this section, we present an anonymous authentication scheme for RFID systems that employs anonymizers and has several appealing features that are important for practical applications:

Anonymity and untraceability of tags against readers. Our scheme allows tags to authenticate to readers without revealing any information that can be used to identify or trace a tag. Hence, even adversaries that can corrupt readers cannot identify or link the transactions of a tag. This is a major improvement to existing RFID systems that usually require the strong assumption of trusted readers (cf. Sections 3.2 and 3.3).

Tag authentication. Our protocol ensures that even adversaries that can corrupt anonymizers and readers cannot impersonate legitimate tags. This is an important advantage compared to existing RFID systems, where a compromised reader usually has a severe impact on the security and privacy of all tags in the system [12, 65, 139].

Availability. In our scheme, the adversary cannot manipulate (i.e., invalidate) legitimate tags without attacking an anonymizer. Availability is a crucial requirement in practice that is often not considered in the design of privacy-preserving RFID systems (cf. Section 3.3).

Efficiency for tags. Our protocol does *not* require tags to perform public-key cryptography and, in contrast to existing solutions to anonymous tag authentication (cf. Section 3.3), matches the computational capabilities of standard RFID tags.

5.4.1 Trust Model and Assumptions

Following [164], we make the following assumptions:

Adversary \mathcal{A} . As in most RFID security models, we assume \mathcal{A} to control the wireless communication channel between readers, anonymizers and tags. This means that \mathcal{A} can eavesdrop, manipulate, delete and reroute all protocol messages sent by \mathcal{R} , \mathcal{P} and \mathcal{T} . Moreover, \mathcal{A} can obtain useful information (e.g., by visual observation) on whether \mathcal{R} accepted \mathcal{T} as a legitimate tag [99, 190].

Issuer \mathcal{I} . We assume \mathcal{I} to be trusted and that \mathcal{I} initializes tags and readers in a secure environment.

Reader \mathcal{R} . We assume that all readers have access to the same information and thus can be subsumed as one single reader entity \mathcal{R} . Moreover, \mathcal{R} can perform public-key cryptography and can handle multiple instances of the anonymous tag authentication protocol with different tags in parallel. In contrast to Section 5.3, we now consider \mathcal{R} to be untrusted, i.e., the adversary can get full control over \mathcal{R} and all the data stored on \mathcal{R} .

Tags \mathcal{T} . Each tag \mathcal{T} is a passive device. This means that a tag cannot initiate communication or participate in more than one protocol-run at the same time, has a narrow communication range (i.e., a few centimeters to meters) and erases its temporary state (i.e., all session-specific information and randomness) after it gets out of the reading range (i.e., the electromagnetic field) of \mathcal{R} or \mathcal{P} .

As recently discussed by Burmester et al. [35], protocols that preserve the privacy of corrupted tags are typically very complex and inefficient and hence, not suitable for most practical RFID applications. Instead, they suggest to frequently revoke and to reissue tags at frequent intervals. In this way, the privacy loss of a tag whose secret has been disclosed is limited to only a small time period. Indeed, this is in line with many use cases like electronic tickets, where tags are expected to expire after some time. Moreover, in practice there are several moderately prized RFID tags that are protected against a variety of physical attacks [9, 141]. Further, as we discuss later in Chapters 6 to 9, emerging hardware-based security primitives such as Physically Unclonable Functions (PUFs) enable physical tamper-protection also for low-cost RFID tags [184, 24, 54]. Hence, we assume \mathcal{T} to be trusted, which means that \mathcal{A} cannot obtain the secrets of \mathcal{T} .

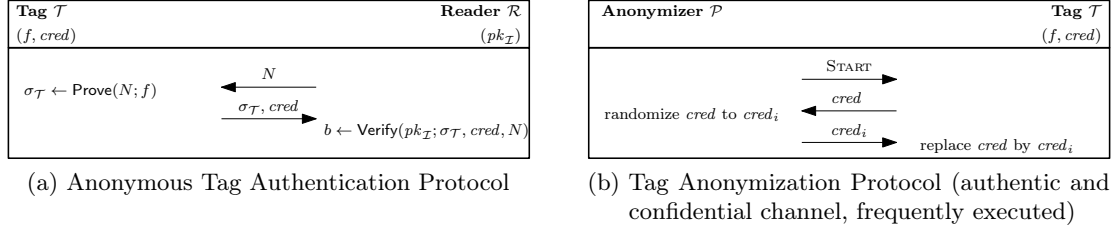


Figure 5.4: Overview of Anonymizer-Enabled Anonymous Authentication for RFID

Anonymizers \mathcal{P} . Anonymizers can perform public-key cryptography and can handle multiple parallel instances of the anonymization protocol with different tags. Similar to readers, we consider anonymizers to be untrusted. Hence, \mathcal{A} can get full control of anonymizers and their secrets. Since a tag \mathcal{T} does not possess the required computational resources to update its state, it can always be tracked between two anonymizations. Therefore, we again assume that each tag \mathcal{T} is frequently anonymized by an honest anonymizer (cf. Section 5.3.1). In practice, this can be achieved by a personal anonymizer that is trusted by its owner, i.e., the tag user.

5.4.2 Protocol Specification

Each tag \mathcal{T} is initialized by the tag issuer \mathcal{I} with a tag-specific authentication secret f and a corresponding anonymous credential $cred$. In the tag authentication protocol (cf. Figure 5.4a), the reader \mathcal{R} challenges \mathcal{T} to sign a random message N . \mathcal{T} returns $cred$ and a partial signature $\sigma_{\mathcal{T}}$ on N , which can be verified using the public key $pk_{\mathcal{I}}$ of \mathcal{I} . If the verification succeeds, \mathcal{R} has assurance that $\sigma_{\mathcal{T}}$ has been created by a tag that has been initialized by \mathcal{I} . Hereby, the structure of $cred$ and $\sigma_{\mathcal{T}}$ ensures that (1) only \mathcal{I} can create a valid $cred$ for any secret f , (2) only a tag that has been initialized by \mathcal{I} can create a valid $\sigma_{\mathcal{T}}$ that can be verified with regard to $cred$ and $pk_{\mathcal{I}}$, and (3) \mathcal{R} does not learn any information that allows \mathcal{R} to deduce the identity of \mathcal{T} .

Since $cred$ is included in each partial signature issued by \mathcal{T} , $cred$ could be used as an identifier of \mathcal{T} . This would allow linking all partial signatures $\sigma_{\mathcal{T}}$ issued by \mathcal{T} and to trace \mathcal{T} . Hence, to provide untraceability of tags, it is crucial that each partial signature $\sigma_{\mathcal{T}}$ issued by \mathcal{T} contains a different $cred$. The construction of $cred$ allows to transform (re-randomize) $cred$ into different anonymous credentials $cred_1, cred_2, \dots$ for the same secret f without knowing the secret key of \mathcal{I} . However, since this transformation requires

public-key operations (i.e., exponentiations) it cannot be performed by \mathcal{T} . Hence, \mathcal{T} must frequently engage the tag anonymization protocol (cf. Figure 5.4b) with \mathcal{P} , which re-randomizes $cred$ for \mathcal{T} .

In our scheme we adapt the anonymous credential system by Chen et al. [42] for our purpose. This scheme is very promising with regard to anonymizer-enabled RFID systems since it allows to split the signature generation between a constrained device and one with higher capabilities. However, due to the limited computational capabilities of RFID tags, it cannot be applied directly. Hence, we removed the support for user-controlled anonymity. This means that, our protocol always ensures the unlinkability of all partial signatures issued by a user (e.g., a tag), whereas the scheme by Chen et al. [42] allows the user to decide to what extent partial signatures can be linked. Moreover, the signing protocol by Chen et al. [42] requires the signer to perform exponentiations, which exceeds the capabilities of most RFID tags in practice. Hence, we employ a similar time-memory tradeoff as used by Liu et al. [122]. This means that a part of the exponentiation is pre-computed by \mathcal{I} and stored on the tag during the tag initialization and, instead of performing the exponentiation, the tag only needs to compute a few multiplications using the pre-computed values in its memory.

The main security objective of our protocol is anonymous tag authentication. More precisely, \mathcal{R} should *only* accept legitimate tags *without* being able to link their transactions (anonymity and unlinkability of tags against readers).

There are three setup protocols where the reader \mathcal{R} , anonymizer \mathcal{P} and the tag \mathcal{T} are initialized and their system parameters (e.g., keys) are generated and defined. A protocol between \mathcal{T} and \mathcal{P} ensures anonymity and unlinkability of tags whereas a second protocol between \mathcal{T} and \mathcal{R} covers anonymous tag-to-reader authentication. Moreover, there is an algorithm to revoke tags and anonymizers, respectively.

System Initialization

Reader setup: $\text{Init}(1^l) \rightarrow (sk_{\mathcal{I}}, pk_{\mathcal{I}}, \text{RL})$. Given a security parameter $l = (q, \kappa, n) \in \mathbb{N}^3$, the tag issuer \mathcal{I} generates the secret parameters $sk_{\mathcal{I}}$ of issuer \mathcal{I} and the corresponding public system parameters $pk_{\mathcal{I}}$. \mathcal{I} generates an admissible pairing $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, P_1, P_2, e) \leftarrow \text{GenPair}(1^q)$ (Definition 2.1), chooses two secret parameters $x, y \xleftarrow{\$} \mathbb{Z}_q$ and computes $X \leftarrow xP_2$ and $Y \leftarrow yP_2$ in \mathbb{G}_2 . Then, \mathcal{I} chooses a random collision-resistant one-way hash function $\text{Hash} : \{0, 1\}^* \rightarrow \{0, 1\}^\kappa$ (Definitions 2.6 and 2.7) and initializes

the secret database $\text{DB} \leftarrow \{\}$ and the revocation list $\text{RL} \leftarrow \{\}$. The secret key of \mathcal{I} is $sk_{\mathcal{I}} \leftarrow (x, y, \text{DB})$ and the corresponding public system parameters are $pk_{\mathcal{I}} \leftarrow (l, q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, P_1, P_2, e, X, Y, \text{Hash}, \text{RL})$.

Anonymizer setup: $\text{SetupAnon}(\mathcal{P}_i) \rightarrow K_i$. The issuer \mathcal{I} checks if \mathcal{P}_i has already been initialized, i.e., if there is a $(\mathcal{P}_i, K_i) \in \text{DB}$ for some K_i . If this is the case, \mathcal{I} aborts. Otherwise \mathcal{I} generates a symmetric encryption key $K_i \leftarrow \text{Genkey}(1^s)$, adds (\mathcal{P}_i, K_i) to DB and initializes \mathcal{P}_i with K_i .

Tag setup: $\text{SetupTag}(\mathcal{P}_i, \mathcal{T}_j, sk_{\mathcal{I}}) \rightarrow S_j$. The issuer \mathcal{I} first checks that \mathcal{P}_i has been initialized but has not been blacklisted. Moreover, \mathcal{I} checks that \mathcal{T}_j has not already been initialized, i.e., that there is no $(\mathcal{T}_j, S_j, \mathcal{P}_i) \in \text{DB}$ (for some S_j and some \mathcal{P}_i). If one of these checks fails, \mathcal{I} aborts. Otherwise \mathcal{I} generates a secret (signing) key f and a corresponding anonymous credential $cred = (D, E, F)$ for \mathcal{T} . Moreover, \mathcal{I} pre-computes \mathcal{G} , t and h that are used later by \mathcal{T}_j in the tag authentication protocol to reduce the number of computations to be performed by \mathcal{T}_j . Therefore, \mathcal{I} chooses $f, r \xleftarrow{\$} \mathbb{Z}_q$ and computes $D \leftarrow rP_1$, $E \leftarrow yD$, $F \leftarrow (x + xyf)D$, $\beta \leftarrow e(E, X)$, $\mathcal{G} \leftarrow \{\beta_0, \dots, \beta_{q-1}\}$ where $\beta_k = \beta^{2^k}$, $t \leftarrow 1$, $h \leftarrow \text{Hash}(D, E, F)$ and $S_j \leftarrow (f, K_i, \mathcal{G}, D, E, F, t, h)$. Finally, \mathcal{I} adds $(\mathcal{T}_j, S_j, \mathcal{P}_i)$ to DB and initializes \mathcal{T}_j with S_j .

Tag Authentication Protocol

$\text{AuthTag}[\mathcal{T}_j : S_j ; \mathcal{R} : \text{RL} ; * : pk_{\mathcal{I}}] \rightarrow [\mathcal{T} : - ; \mathcal{R} : out_{\mathcal{R}}]$. The tag authentication protocol is shown in Figure 5.5. In this protocol, a tag \mathcal{T}_j anonymously authenticates to the reader \mathcal{R} . Therefore, \mathcal{R} challenges \mathcal{T}_j to sign a random challenge N_r . Upon receipt of N_r , \mathcal{T}_j computes a signature of knowledge $\sigma \leftarrow (D, E, F, v, s)$ (that includes the credential $cred = (D, E, F)$ of \mathcal{T}_j) in a similar way as in the protocol by Chen et al. [42]. However, in our case, \mathcal{T}_j uses the time-memory tradeoff by Liu et al. [122] to compute $\tau \leftarrow \beta^{t \cdot z'}$ for $z' \xleftarrow{\$} \mathbb{Z}_q$.⁹ Hereby, t ensures that the precomputed set \mathcal{G} is adjusted to the current randomization of the credential $cred$ of \mathcal{T}_j (see the tag anonymization protocol that is explained further below). Upon receipt of $\sigma = (D, E, F, v, s)$, \mathcal{R} verifies

⁹Consider the square-and-multiply algorithm (SQM), which is a standard algorithm for fast modular exponentiation. Note that the set \mathcal{G} contains the precomputed results of the squaring operations performed by the SQM. Hence, \mathcal{T}_j only needs to perform the multiplications of the SQM, which significantly reduces the computational complexity of the exponentiation for \mathcal{T}_j .

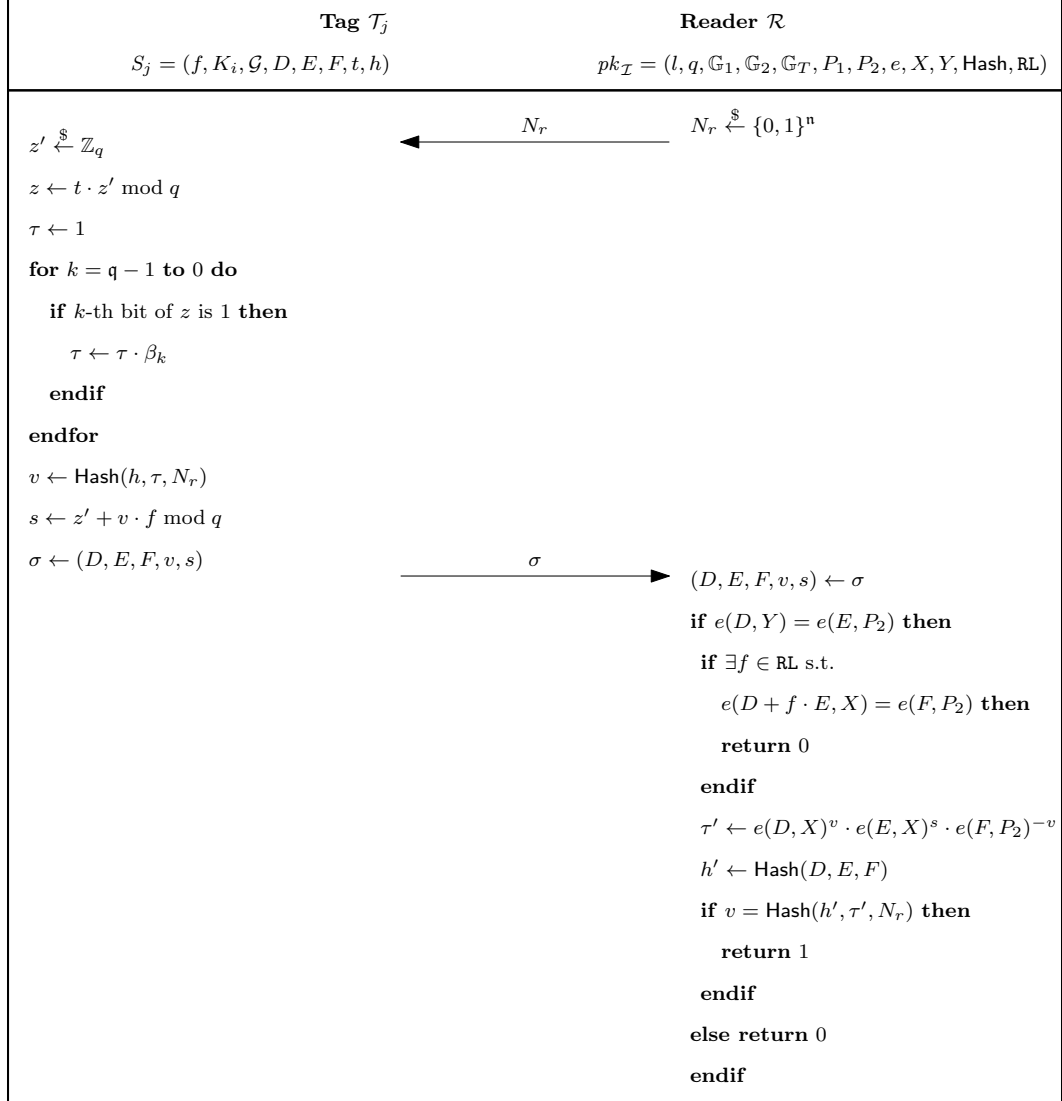


Figure 5.5: Anonymous Tag Authentication Protocol

that (1) (D, E, F) is a valid credential $cred$ with regard to $pk_{\mathcal{I}}$, (2) the secret f which corresponds to (D, E, F) has not been revoked (i.e., \mathcal{T}_j has not been added to RL) and (3) (v, s) is a valid signature of knowledge on N_r with regard to (D, E, F) and $pk_{\mathcal{I}}$. If the verification is successful, \mathcal{R} accepts \mathcal{T}_j as a legitimate tag and returns 1. Otherwise \mathcal{R} returns 0.

Tag Anonymization Protocol

AnonTag $[\mathcal{T}_j : S_j ; \mathcal{P}_i : K_i ; * : pk_{\mathcal{I}}] \rightarrow [\mathcal{T}_j : S'_j ; \mathcal{P}_i : -]$. In this protocol, an anonymizer \mathcal{P}_i updates the credential $cred = (D, E, F)$ and the precomputed values (t, h) of the tag \mathcal{T}_j that are later used by \mathcal{T}_j in the tag authentication protocol. Hereby, \mathcal{P}_i and \mathcal{T}_j communicate over an authentic and confidential channel based on symmetric encryption. Therefore, K_i , N_i and N_j are used to encrypt the communication between \mathcal{P}_i and \mathcal{T}_j and to mutually authenticate both parties. In the second protocol message, \mathcal{T}_j sends its credential $cred = (D, E, F)$ for f to \mathcal{P}_i , which re-randomizes it to another credential $cred^* = (D^*, E^*, F^*)$ for f that can still be verified by the public key $pk_{\mathcal{I}}$ of \mathcal{I} . Finally, \mathcal{T}_j replaces its old credential $cred$ by $cred^*$ and updates h and t such that in the tag authentication protocol \mathcal{T}_j can adjust \mathcal{G} to the new credential $cred^*$ (see the tag authentication protocol explained in the previous paragraph). The tag anonymization protocol is detailed in Figure 5.6.

Tag and Anonymizer Revocation

Tag revocation: $\text{RevTag}(\mathcal{T}_j, S_j) \rightarrow \text{RL}$. To revoke a tag \mathcal{T}_j , the tag issuer \mathcal{I} first checks if there is a $(\mathcal{T}_j, S_j, \mathcal{P}_i) \in \text{DB}$ for some $S_j = (f, K_i, \mathcal{G}, D, E, F, t, h)$ and some \mathcal{P}_i . If this is the case, \mathcal{I} adds (\mathcal{T}_j, f) to the revocation list RL and sends RL to \mathcal{R} using an authentic channel.

Anonymizer revocation: $\text{RevAnon}(\mathcal{P}_i)$. To revoke an anonymizer \mathcal{P}_i , the tag issuer \mathcal{I} first checks if there is a $(\mathcal{P}_i, K_i) \in \text{DB}$ for some K_i . If this is the case, \mathcal{I} blacklists \mathcal{P}_i and removes (\mathcal{P}_i, K_i) from DB .¹⁰

¹⁰The revocation of an anonymizer means that all deployed tags that are assigned to this anonymizer become traceable because they are no longer anonymized but they can still authenticate to the readers. Further, in case the revoked anonymizer has been compromised, these tags may be at risk of being invalidated because a malicious anonymizer could write faked data to them, which prevents these tags from successfully authenticating to readers. Hence, all already deployed tags assigned to the

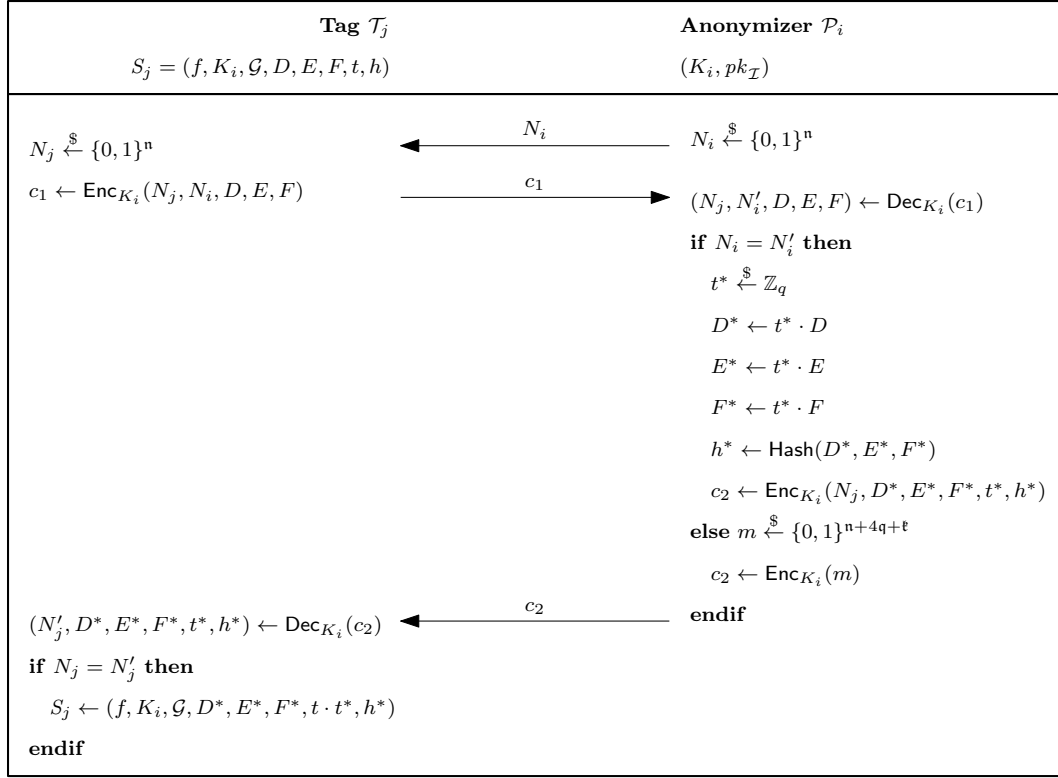


Figure 5.6: Tag Anonymization Protocol for Anonymous Authentication

5.4.3 Performance Evaluation

Note that the tag user does not notice the interaction between the anonymizer and a tag, whereas tag authentication usually requires the user to wait (e.g., at a door or gate) until the authentication protocol completes. Thus, most practical applications have strict time constraints on the identification protocol [178, 166] while there are no critical constraints on the tag anonymization protocol. Moreover, compared to a tag, the reader possesses much more computing power. Hence, in this section, we only focus on the resources required by the tag to execute the tag authentication protocol. In particular, we consider the computational, communication and memory effort of the tag.

Computation. The tag authentication protocol requires the tag to generate q random bits, to perform two multiplications and one addition in \mathbb{Z}_q , $(q - 1)/2$ multiplications

revoked anonymizer should be re-issued.

in \mathbb{G}_T (on average) and one hash digest. Compared to a plain exponentiation, the time-memory tradeoff by Liu et al. [122] saves q squarings in \mathbb{G}_T , which are precomputed by the issuer \mathcal{I} and stored in the memory of the tag when the tag is initialized. Note that these pre-computed values can be reused in each tag authentication protocol run. To achieve a security level that is comparable to RSA 1,024 bit, a reasonable choice for the security parameters is $q = 154$ [25], $n = 128$ and $\ell = 160$ [1].

Communication. The tag authentication protocol requires to send an n bit random value from \mathcal{R} to \mathcal{T} and three elements of \mathbb{G}_1 , one ℓ bit hash digest and one element of \mathbb{Z}_q from \mathcal{T} to \mathcal{R} . Hence, the total communication complexity of the tag authentication protocol is $n + 3l_{\mathbb{G}_1} + \ell + q$ bits, where $l_{\mathbb{G}_1}$ is the size (in bits) of an element of \mathbb{G}_1 . For the choice of parameters discussed above $l_{\mathbb{G}_1} = 308$, which means that the total communication complexity of the tag authentication protocol is 1,366 bits.

Memory. Each tag must store two elements of \mathbb{Z}_q , one s -bit key, q elements of \mathbb{G}_T , three elements of \mathbb{G}_1 and one n -bit hash digest. This means that each tag must store $2q + n + q \cdot l_{\mathbb{G}_T} + 3l_{\mathbb{G}_1} + \ell$ bits in total, where $l_{\mathbb{G}_T}$ is the size (in bits) of an element of \mathbb{G}_T . For the parameter choices discussed above $l_{\mathbb{G}_T} = 923$, which means that each tag must store about 17.6 KByte of data.

Although we solved the problem of reducing the computational costs for the RFID tags to match the capabilities of existing tags, the memory requirements of our scheme still need further optimization.¹¹

5.4.4 Security Analysis

Now we formally define and prove tag authentication and unlinkability of tags for our protocol. In contrast to the case of trusted readers, where we could built on top of the V-Model, there is no existing RFID security and privacy framework that considers untrusted readers and anonymizers. Hence, we specify our security goals of tag authentication and unlinkability of tags using game-based definitions (which is a common approach in cryptography) that are based on the definitions of the V-Model and the security definitions by Chen et al. [42].

¹¹Most currently available RFID tags in practice (like MiFare Plus [141] that are used in electronic ticketing systems) can perform symmetric encryption (DES, 3DES, AES), keyed hashing based on encryption, generate random numbers and provide up to 8 KByte of memory.

Tag Authentication

Tag authentication means that an adversary \mathcal{A} should not be able to make an honest reader \mathcal{R} to accept \mathcal{A} as a legitimate tag. We formalize tag authentication by a security experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathcal{T}\text{-aut}} = \text{out}_{\mathcal{R}}^{\pi}$, where a p.p.t. adversary \mathcal{A} must make an honest \mathcal{R} to authenticate \mathcal{A} as some legitimate tag \mathcal{T}_j by returning $\text{out}_{\mathcal{R}}^{\pi} = 1$ in some instance π of the tag authentication protocol **AuthTag**. Hereby, \mathcal{A} can arbitrarily interact with the RFID system. However, since in general it is not possible to prevent simple relay attacks, \mathcal{A} is not allowed to just forward all messages from \mathcal{T}_j to \mathcal{R} in instance π .¹² This means that at least some of the protocol messages that made \mathcal{R} to accept must have been (partly) computed by \mathcal{A} without knowing the secrets of \mathcal{T}_j .

Definition 5.6 (Tag Authentication). *An anonymous RFID system achieves tag authentication if for every p.p.t. adversary \mathcal{A} $\Pr [\mathbf{Exp}_{\mathcal{A}}^{\mathcal{T}\text{-aut}} = 1]$ is negligible.*

Unlinkability of Tags

Unlinkability means that an adversary \mathcal{A} cannot distinguish tags based on their communication.¹³ This means that the protocol messages generated by tags should not leak any information to \mathcal{A} that allows \mathcal{A} to identify or trace them. We formalize tag authentication by a security experiment $\mathbf{Exp}_{\mathcal{A}}^{\text{prv-}b}$ for $b \xleftarrow{\$} \{0, 1\}$, where a p.p.t. adversary \mathcal{A} interacts with an oracle \mathcal{O}_b that either represents two identical ($b = 0$) or two different ($b = 1$) legitimate tags \mathcal{T}_0 and \mathcal{T}_1 . Hereby, \mathcal{A} can arbitrarily interact with the RFID system and \mathcal{O}_b . However, to exclude trivial attacks (e.g., denial-of-service attacks), \mathcal{A} is not allowed to corrupt an anonymizer nor to disturb the anonymization protocol (cf. Section 5.4.1). Finally \mathcal{A} returns a bit b' .

Definition 5.7 (Unlinkability). *An RFID system achieves unlinkability of tags if for every p.p.t. adversary \mathcal{A} $\mathbf{Adv}_{\mathcal{A}}^{\text{prv}} = |\Pr [\mathbf{Exp}_{\mathcal{A}}^{\text{prv-}0} = 1] - \Pr [\mathbf{Exp}_{\mathcal{A}}^{\text{prv-}1} = 1]|$ is negligible.*

Security Proof

We are now ready to formally state the security and privacy properties of the protocol presented in Section 5.4.2 in the following theorem:

¹²Note that relay attacks can be mitigated by distance bounding techniques. However, for simplicity we excluded relay attacks because our focus is anonymous authentication against malicious readers.

¹³Unlinkability implies anonymity since an adversary who can identify tags can also trace them.

Theorem 5.2 (Tag Authentication and Unlinkability). *The RFID scheme described in Section 5.4.2 achieves tag authentication (Definition 5.6) under the Bilinear LRSW Assumption (Definition 2.3) if the underlying hash function is collision-resistant (Definition 2.7). Further, it achieves unlinkability (Definition 5.7) under the Decisional Diffie-Hellman Assumption in \mathbb{G}_1 (Definition 2.2) if the underlying encryption scheme is real-or-random indistinguishable (Definition 2.11) and the underlying hash function is collision-resistant (Definition 2.7).*

Proof of tag authentication (Theorem 5.2). Assume by contradiction that \mathcal{A} is an adversary such that $\Pr[\mathbf{Exp}_{\mathcal{A}}^{\mathcal{T}\text{-aut}} = 1]$ is non-negligible. In the following, we show how to use \mathcal{A} to construct an adversary that either violates the Bilinear LRSW Assumption (Definition 2.3) or the collision-resistance of the underlying hash function (Definition 2.7).

Note that $\mathbf{Exp}_{\mathcal{A}}^{\mathcal{T}\text{-aut}} = 1$ implies that \mathcal{A} computed some protocol message (D, E, F, v, s) for a given reader challenge N_r such that $e(D, Y) = e(E, P_2)$ and $v = \text{Hash}(h, \tau, N_r)$ where $h = \text{Hash}(D, E, F)$ and $\tau = e(D, X)^v \cdot e(E, X)^s \cdot e(F, P_2)^{-v}$. Hereby, \mathcal{A} has two possibilities: (1) reuse a credential (D, E, F) from a previous tag authentication protocol run or (2) create a new (forged) credential (D, E, F) . In the following, we show that if \mathcal{A} is successful in the first case, then \mathcal{A} can be used to find a collision of Hash , which contradicts the assumption that Hash is collision-resistant (Definition 2.7). Moreover, if \mathcal{A} is successful in the second case, then \mathcal{A} can be used to violate the Bilinear LRSW Assumption (Definition 2.3). Hence, the collision-resistance property of Hash and the Bilinear LRSW Assumption ensure that $\Pr[\mathbf{Exp}_{\mathcal{A}}^{\mathcal{T}\text{-aut}} = 1]$ is negligible.

CASE 1: \mathcal{A} reuses an old credential. Assume by contradiction that \mathcal{A} uses (a randomized version of) a credential (D', E', F') from a previous transcript $(N'_r, (D', E', F', v', s'))$ of the tag authentication protocol to forge a signature (v, s) on a new reader challenge N_r . Note that $\Pr[N_r = N'_r]$ is negligible since N_r is uniformly chosen in each tag authentication protocol-run. Hence, if \mathcal{R} accepts an old signature (v', s') for a new challenge N_r , then with overwhelming probability $v' = \text{Hash}(h', \tau', N'_r) = \text{Hash}(h', \tau', N_r)$ and $N_r \neq N'_r$. This means that \mathcal{A} found a collision of Hash . However, since Hash is assumed to be collision-resistant, this can only happen with negligible probability. Therefore, \mathcal{A} must have computed a new signature of knowledge (v, s) such that $v = \text{Hash}(h', \tau, N_r)$ and $s = z' + v \cdot f \bmod q$ where $\tau = e(E', X)^{z' \cdot t}$. Note that (v, s) includes a proof of knowledge of a value f such that $e(D' + f \cdot E', X) = e(F', P_2)$, which is a standard Σ -protocol for proving knowledge of a discrete logarithm. It follows from the proof-of-knowledge

property that, if \mathcal{A} can compute a valid (v, s) , then there is a p.p.t. algorithm (knowledge extractor) that can extract f from \mathcal{A} . This implies that \mathcal{A} knows f . However, \mathcal{A} can guess f only with negligible probability. Hence, the proof-of-knowledge property ensures that \mathcal{A} can forge a signature (v, s) on a message N_r for a given credential (D, E, F) only with negligible probability.

CASE 2: \mathcal{A} creates a new credential. Assume that \mathcal{A} can construct a tuple (D, E, F, v, s) where (D, E, F) is *not* (a randomized version of) a credential from a previous tag authentication protocol. In the following, we show that \mathcal{A} can be used to construct an adversary $\mathcal{A}_{\text{bLRSW}}$ against the Bilinear LRSW Assumption (Definition 2.3). Given access to oracle $\mathcal{O}_{X,Y}$ and the public parameters $pk_{\text{bLRSW}} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, P_1, P_2, e, X, Y)$, $\mathcal{A}_{\text{bLRSW}}$ simulates the initialization algorithm `Init` of the RFID system to \mathcal{A} as specified in Section 5.4.2 but uses pk_{bLRSW} to construct $pk_{\mathcal{I}}$. Note that $\mathcal{A}_{\text{bLRSW}}$ does *not* know the secret parameters (x, y) of the simulation of the RFID system, which are required for the simulation of the `SetupTag` algorithm. However, $\mathcal{A}_{\text{bLRSW}}$ can simulate `SetupTag` with the help of $\mathcal{O}_{X,Y}$. Instead of using (x, y) to compute the credential for the tag to be initialized, $\mathcal{A}_{\text{bLRSW}}$ chooses $f \xleftarrow{\$} \mathbb{Z}_q$ and queries $\mathcal{O}_{X,Y}(f)$, which responds with a tuple $(D, yD, (x + fxy)D)$. Note that by definition of $\mathcal{O}_{X,Y}$ $D \xleftarrow{\$} \mathbb{G}_1$, which means that D can be expressed as $D = rP_1$ where $r \xleftarrow{\$} \mathbb{Z}_q$. Therefore, the output generated by $\mathcal{O}_{X,Y}$ is a valid credential and hence, the simulation of `SetupTag` is perfect. Moreover, $\mathcal{A}_{\text{bLRSW}}$ can perfectly simulate all other algorithms and protocols of the RFID system since they do not require knowledge of (x, y) . Thus, after a polynomial number of queries to $\mathcal{A}_{\text{bLRSW}}$, \mathcal{A} returns a protocol message (D, E, F, v, s) for a given N_r that makes \mathcal{R} to return $\text{out}_{\mathcal{R}} = 1$. Since (v, s) includes a proof of knowledge of a value f such that $e(D + f \cdot E, X) = e(F, P_2)$, $\mathcal{A}_{\text{bLRSW}}$ can use the corresponding knowledge extractor to extract f from \mathcal{A} . Finally, $\mathcal{A}_{\text{bLRSW}}$ returns a tuple (f, D, E, F) . Since (D, E, F) is not (a randomized version of) a credential from a previous tag authentication protocol, it holds that $\mathcal{O}_{X,Y}$ has never been queried for the corresponding secret f . Hence, (f, D, E, F) is a valid solution to the Bilinear LRSW problem, which is a contradiction to the Bilinear LRSW Assumption (Definition 2.3). Hence, \mathcal{A} can generate a valid tuple (D, E, F, v, s) for a given message N_r that is not based on an existing credential only with negligible probability. \square

Proof of unlinkability (Theorem 5.2). Recall that unlinkability (Definition 5.7) requires that \mathcal{A} cannot distinguish whether \mathcal{O}_b represents two identical or two different tags.

We show that if \mathcal{A} has non-negligible advantage $\mathbf{Adv}_{\mathcal{A}}^{\text{prv}}$, then \mathcal{A} can be used to break the DDH-Assumption in \mathbb{G}_1 (Definition 2.2) or the real-or-random indistinguishability of the encryption scheme (Definition 2.11). For this purpose, we show that (1) the communication between tags and anonymizers does not leak any information that helps \mathcal{A} to distinguish, and (2) that executions of the tag authentication protocol cannot be linked. For the first claim, we show that $\mathbf{Adv}_{\mathcal{A}}^{\text{prv}}$ of \mathcal{A} does not change whether \mathcal{A} can eavesdrop executions of the tag anonymization protocol **AnonTag** or not. Here, we use the standard approach of game hopping. Let \mathbf{G}_0 be the scenario, where \mathcal{A} interacts with the real RFID system, i.e., consider a hypothetical simulator \mathcal{C}_0 that honestly simulates the whole RFID system to \mathcal{A} . Obviously, \mathcal{A} has advantage $\mathbf{Adv}_{\mathcal{A}}^{\text{prv}}$ in this case.

Next, we consider the game \mathbf{G}_1 that is played by a simulator \mathcal{C}_1 , which behaves exactly as \mathcal{C}_0 with the following difference: Whenever a tag \mathcal{T}_j runs the tag anonymization protocol with some anonymizer \mathcal{P}_i , \mathcal{C}_1 replaces the ciphertexts c_1 and c_2 by dummy encryptions c'_1 and c'_2 that are constructed as explained further below. Observe that \mathcal{C}_1 ensures that \mathcal{T}_j and \mathcal{P}_i perform the same computations as if they received the correct ciphertexts c_1 and c_2 . The encryptions c'_1 and c'_2 are generated as follows: In parallel to the execution of the anonymization protocol between \mathcal{T}_j and \mathcal{P}_i , a second instance of the tag anonymization protocol is honestly executed between some other tag $\mathcal{T}_{j'}$ where $j' \neq j$ and an anonymizer $\mathcal{P}_{i'}$ (which can be equal to \mathcal{P}_i). The dummy encryptions c'_1 and c'_2 occurring in this second protocol-run are used as a replacement for c_1 and c_2 . At the end of this second protocol-run, the involved tag is reset to its state before the protocol execution.¹⁴ This ensures that encryptions of only well-formed plaintexts are transmitted. The only difference from an attacker's point of view is that in \mathbf{G}_0 the correct (or real) plaintexts are encrypted, while in \mathbf{G}_1 only randomly chosen (but well-formed) plaintexts are encrypted. If the advantage $\mathbf{Adv}_{\mathcal{A}}^{\text{prv}}$ of \mathcal{A} significantly differs in \mathbf{G}_0 and \mathbf{G}_1 , then \mathcal{A} can be turned into a real-or-random distinguisher for the underlying encryption scheme (cf. Definition 2.11). Thus, since the encryption scheme is assumed to be real-or-random indistinguishable, the difference of the advantage of \mathcal{A} in \mathbf{G}_0 and \mathbf{G}_1 is negligible.

Finally, we define the game \mathbf{G}_2 to be as \mathbf{G}_1 with the only difference that \mathcal{A} is not allowed to see the messages exchanged in any instance of the tag anonymization protocol. Since the dummy encryptions in \mathbf{G}_1 are by definition *independent* of the computations and values used in the tag anonymization protocol and since the random value N_i is *not* used

¹⁴Alternatively, we can consider a pair of tag and anonymizer that is created outside the system, i.e., that are never reported to \mathcal{A} and that are only used for generating dummy ciphertexts.

in the computations that update the tag state S_j , the adversary does not gain any useful information by eavesdropping on the tag anonymization protocol. More precisely, any adversary in \mathbb{G}_2 can be easily turned into an adversary in \mathbb{G}_1 and vice versa by adding or removing dummy encryptions. Thus, the maximum possible advantage for linking tags is the same in \mathbb{G}_1 and \mathbb{G}_2 . In particular, the communication between tags and anonymizers does not leak any useful information, which proves the first claim.

Now we show that executions of the anonymous tag authentication protocol **AuthTag** cannot be linked. With $\sigma[f, cred(f)]$ we denote a signature σ that has been generated by \mathcal{O}_b using the secret signing key f and the credential $cred(f)$ for f . Let f_0 be the signing key of \mathcal{T}_0 and f_1 be the key of \mathcal{T}_1 . Note that both \mathcal{T}_0 and \mathcal{T}_1 are simulated by \mathcal{O}_b . In the following, we show that the distributions $\delta := \langle \sigma_0[f_0, cred(f_0)], \sigma_1[f_0, cred(f_0)] \rangle$ and $\delta' := \langle \sigma_2[f_0, cred(f_0)], \sigma_3[f_1, cred(f_1)] \rangle$ are computationally indistinguishable. More precisely, we show that if \mathcal{A} can distinguish between δ and δ' with non-negligible advantage $\mathbf{Adv}_{\mathcal{A}}^{\text{prv}}$, then \mathcal{A} can be used to construct an algorithm \mathcal{A}_{DDH} that violates the DDH-Assumption in \mathbb{G}_1 (Definition 2.2).

Let (D_i, E_i, F_i) be the credential used to compute a signature σ_i . Note that all credentials (D_i, E_i, F_i) for $i \in \{0, 1, 2\}$ are randomizations of the credential $cred(f_0)$. Hence, $F_i = \alpha D_i$ for $i \in \{0, 1\}$ and $\alpha = x + xyf_0$. Moreover, for all signature pairs distributed according to δ there is a $\gamma \in \mathbb{Z}$ such that $D_1 = \gamma D_0$. Similarly, all credentials (D_3, E_3, F_3) are randomized versions of $cred(f_1)$ and $F_3 = \alpha' D_3$ for $\alpha' = x + xyf_1$. Further, for all signature pairs distributed according to δ' there is a $\gamma' \in \mathbb{Z}$ such that $D_3 = \gamma' D_2$. Note that for all signature pairs distributed according to δ it holds that $(F_0, D_1, F_1) = (\alpha D_0, \gamma D_0, \alpha \gamma D_0)$ is a DDH-tuple, while this is *not* true for the signatures $(F_2, D_3, F_3) = (\alpha D_2, \gamma' D_2, \alpha' \gamma' D_2)$ in δ' . However, the DDH-Assumption in \mathbb{G}_1 (Definition 2.2) ensures that both distributions δ and δ' are computationally indistinguishable. Hence, \mathcal{A} cannot link tags based on their communication in the tag authentication protocol, which finishes the proof. \square

5.5 Conclusion

In this chapter, we presented two RFID systems that enable cost-effective privacy-preserving authentication and anonymous authentication, respectively, of RFID tags to readers. Both protocols use anonymizers, which are separate devices specifically designated to ensure the privacy of tags.

The first protocol achieves narrow-strong privacy without requiring tags to perform public-key operations, thus providing a satisfying notion of privacy for low-cost tags in response to an open question raised by Vaudenay [190]. To prove the security of this protocol we introduced a security and privacy model for anonymizer-enabled RFID systems that builds on top of Vaudenay’s model and is backwards compatible with it.

The second protocol enables RFID tags to authenticate to readers without disclosing any information that allows the identification or tracking of tags by readers. This is often sufficient or may be even required by privacy regulations or laws for many RFID-based access control systems such as electronic tickets in practice. As a first step, we solved the problem of reducing the computational costs for the RFID tags to match the capabilities of current mid-range tags. However, the memory requirements of the anonymous tag authentication scheme still need further optimization. Moreover, the protocol does not capture the protection against cloning of tags. A promising approach to solve this open problem with minimal overhead on the tag side are Physically Unclonable Functions (PUFs), which are the focus of the following chapters.

6 Background on Physically Unclonable Functions

Physically Unclonable Functions (PUFs) are increasingly proposed as central building blocks in cryptographic protocols and higher level security architectures. Among others, PUFs enable unique device identification and authentication [156, 184, 148, 167], binding software to hardware platforms [76, 113, 77, 59] and secure storage of cryptographic secrets [195, 120]. Furthermore, they can be integrated into cryptographic algorithms [4] and remote attestation protocols [171]. Today, PUF-based security products are already announced for the market, mainly targeting IP-protection and anti-counterfeiting applications as well as RFID systems [192, 94].

Remark. Parts of this chapter have been published in [3] and [85].

6.1 PUF Concept, Properties and Assumptions

A physically unclonable function (PUF) is a noisy function that is embedded into a physical object, such as an integrated circuit [151, 125, 3]. When queried with a *challenge* x , a PUF generates a *response* $y \leftarrow \text{PUF}(x)$ that depends on both x and the unique device-specific intrinsic physical properties of the object containing the PUF. Since PUFs are subject to noise induced by environmental variations, such as supply voltage and ambient temperature variations, they return slightly different responses when queried with the same challenge multiple times.

PUFs are typically assumed to be *robust*, *physically unclonable*, *unpredictable* and *tamper-evident* and several approaches to quantify and formally define their properties have been proposed (see [3] for an overview). Informally, robustness means that, when queried with the same challenge multiple times, the PUF returns a similar response with high probability. Physical unclonability demands that it is infeasible to produce two PUFs that cannot be distinguished based on their challenge/response behavior. Unpredictability requires that it is infeasible to predict the PUF response to an unknown challenge, even if the PUF can be adaptively queried for a certain number of times. Fi-

nally, a PUF is tamper-evident if any attempt to physically access the PUF irreversibly changes its challenge/response behavior.

In contrast to most cryptographic primitives, whose security can be related to well established (albeit unproven) assumptions, the security of PUFs is assumed to rely on physical properties. The security properties of PUFs can either be evaluated theoretically, based on mathematical models of the underlying physics [187, 194, 193] or experimentally by analyzing PUF instances built in hardware [92, 186, 79, 88, 188]. The first approach has the apparent drawback that mathematical models never capture physical reality in its full extent, which means that the conclusions on PUF security drawn by this approach are naturally debatable. The main drawback of the experimental approach is its limited reproducibility and openness. Even though experimental results have been reported in the literature for some PUF implementations, it is difficult to compare them due to varying test conditions and different analysis methods. Furthermore, raw PUF data is rarely available for subsequent research, which greatly hinders a fair comparison. We present a large-scale security analysis of ASIC implementations of the five most popular intrinsic electronic PUF types in Chapter 7. An independent evaluation of electronic PUFs on ASIC using a simpler methodology for unpredictability has been presented by Bhargava et al. [19].

The security analysis of PUFs is further complicated by the drawbacks of existing approaches to formalize their security properties. Currently there is no widely accepted security model for PUFs while most PUF security models in the literature are not general enough and exclude certain PUF types (such as in [151, 68]), do not reflect all properties of real PUF implementations (as in [151, 68, 76, 161, 4]) or include security parameters that cannot be determined for real PUF implementations in practice (such as in [161, 4, 33]). Existing literature on PUF-based security mechanisms typically uses idealized PUF models that do *not* reflect *real* PUF implementations but capture the *desired* properties of an *ideal* PUF.

While it is unclear whether an ideal PUF exists, a common evaluation framework for the analysis of real PUF implementations is needed to design secure and practical PUF-based systems. Such a framework should (1) capture the security properties of real PUF implementations according to modern cryptographic standards so that it can be used to assess the security of PUF-based cryptographic schemes and security solutions and (2) it should allow for empirically assessing and quantifying the most important properties of

Enrolment Phase	Reconstruction Phase
<i>Generate helper data $h \dots$</i>	<i>\dots that is later used to recreate K.</i>
$y \leftarrow \text{PUF}(x)$ $(K, h) \leftarrow \text{FEGen}(y)$ Store (x, h)	$y' \leftarrow \text{PUF}(x)$ $K \leftarrow \text{FERep}(y', h)$

Figure 6.1: Concept of Fuzzy Extractors

PUFs, including robustness, physical unclonability and unpredictability. We make a first step into this direction in Chapter 8, where we present a PUF security framework that captures the fundamental properties of real PUF implementations.

6.2 PUF Types

There is a variety of PUF implementations (see [125] for an overview). The most appealing ones for the integration into electronic circuits are *electronic PUFs*, which come in different flavors. *Delay-based PUFs* are based on race conditions or frequency variations in integrated circuits and include Arbiter PUFs [116, 148, 121] and Ring Oscillator PUFs [68, 180, 126]. *Memory-based PUFs* exploit the instability of volatile memory elements, such as SRAM cells [76, 87], flip-flops [124, 188] and latches [179, 113]. Finally, *Coating PUFs* [185] use capacitances of a special dielectric coating applied to the chip implementing the PUF.

6.3 Noise Compensation and Privacy Amplification

Many PUF-based applications require PUF responses to be reliably reproducible while at the same time being unpredictable [4, 125, 3]. However, since PUFs are inherently noisy and their responses are not uniformly random, they are typically combined with *fuzzy extractors* [57, 56]. Fuzzy extractors consist of a *secure sketch*, which is an algorithm that maps similar PUF responses to the same value (noise compensation or error correction) and a randomness extractor that extracts full-entropy bit strings from a partially random source (privacy amplification).

Fuzzy extractors and secure sketches generally work in two phases (cf. Figure 6.1). In the *enrolment phase* some helper data h and a uniform bit string K (that could be used as cryptographic secret) is computed from the PUF response y , which is used later

in the *reconstruction phase* to recover $K = \text{FERep}(y', h)$ from a distorted PUF response $y' = y + e$, where e is the error caused by noise. An important property of fuzzy extractors and secure sketches is that, after observing one single helper data value h , there is still some min-entropy left in y and K , which means that h can be stored and transferred publicly without disclosing the full PUF response y or K [57].

More detailed information on fuzzy extractors and a number of practical instantiations can be found in Sections 9.2.1 and 9.2.4 and in the work by Dodis et al. [57].

6.4 Common PUF-based Applications

This section describes the most common approaches to integrate PUFs into identification and authentication schemes and to use PUFs for secure key generation and storage.

6.4.1 Device Identification and Authentication

The classical application of PUFs is the identification and authentication of physical objects, such as electronic devices. In fact, PUFs have been first proposed in the context of anti-counterfeiting solutions that prevent cloning (i.e., unauthorized copying) of products. There are many proposals to build identification and authentication schemes based on PUFs for various devices. We focus on solutions that are applicable to resource-constrained embedded devices such as RFID systems, which are the focus of this thesis.

One of the first proposals of using PUFs for RFID is by Ranasinghe et al. [156], who propose the manufacturer of a PUF-enabled RFID tag to store a set of challenge/response pairs (CRPs) in a database which can later be used by RFID readers that are connected to this database to identify the tag. The idea is that the reader chooses a challenge from the database, queries the tag and checks whether the database contains a tuple that matches the response received from the tag. One problem of this approach is that CRPs cannot be re-used since this would enable replay attacks and allow tracing of tags. Hence, the number of tag authentications is limited by the number of CRPs in the database. This scheme has been implemented based on Arbiter PUFs on an RFID tag and its security and usability has been analyzed by Devadas et al. [54]. A similar approach based on the physical characteristics of SRAM cells has been proposed by Holcomb et al. [87]. The advantage of SRAM PUFs is that they can be implemented based on the existing SRAM cells of the RFID chip without the need for additional hardware. Another approach to PUF-based authentication by Bolotnyy and Robins [24]

aims to prevent unauthorized tracking of tags. A major drawback of their scheme is that tags can only be authenticated a limited number of times without being re-initialized, which enables denial-of-service attacks.

A privacy-preserving PUF-based authentication scheme has been presented by Gassend et al. [67]. They suggest to equip each tag with a PUF that is used to frequently derive new tag identifiers. Since readers cannot recompute these identifiers, the readers have access to a database that stores a tuple $(ID_1, ID_2, \dots, ID_n)$ for each legitimate tag, where ID_0 is a random tag identifier and $ID_i = \text{PUF}(ID_{i-1})$ for $1 \leq i \leq n$. To authenticate to a reader, the tag first sends its current identifier ID_j and then updates its identity to $ID_{j+1} = \text{PUF}(ID_j)$. The reader then checks whether there is a tuple that contains ID_j in the database. In case the reader finds ID_j , it accepts the tag and invalidates all previous database entries ID_k , where $k \leq j$ to prevent replay attacks. A major drawback of this scheme is that a tag can only be authenticated n times without being re-initialized, which, as the authors mention, allows the adversary to perform denial-of-service attacks.

6.4.2 Secure Key Generation and Storage

PUFs can be used to securely bind secrets (such as cryptographic keys) to the physical characteristics of a device. The concept of PUF-based key storage has been presented by Gassend [66] and later generalized by Bringer et al. [32]. Instead of storing the key in non-volatile memory that is vulnerable to invasive hardware attacks, the key is extracted from the physical properties of the underlying hardware each time it is used. This protects the key against unauthorized readout by invasive hardware attacks, such as probing attacks against non-volatile memory. Moreover, in case a tamper-evident PUF is used, any attempt to physically extract the key from the PUF circuit is assumed to change the challenge/response behavior of the PUF and to securely delete the key bound to the PUF.

Since PUF responses are typically not uniformly random and subject to noise, they cannot be used directly as cryptographic keys. Hence, privacy amplification, which adds additional entropy to the PUF response and error correction techniques must be applied before a PUF response can be used as a cryptographic key. The most common approach to achieve this are fuzzy extractors (cf. Section 6.3).

An essential requirement for the creation of cryptographically secure keys is the ability to generate random numbers. Holcomb et al. [87] propose using instable SRAM cells as

a source for true random numbers. Their evaluation results show that 210 random and uniformly distributed bits can be extracted from 2,048 bits of SRAM.

Tuyls et al. [184] propose to use a PUF-based key storage for the secret authentication key of RFID tags. Since the key is inherently hidden within the physical structure of the PUF, obtaining this key by hardware-related attacks is supposed to be intractable for real-world adversaries [68]. According to Tuyls et al. [184], a PUF-based key storage can be implemented with less than 1,000 gates, which is well within the capabilities of common RFID tags. Several other authentication schemes for RFID exist that use PUF-based key storage to protect against unauthorized tracking of tokens [31, 167] and relay attacks [103].

6.5 Attacks on PUFs and PUF-based Systems

6.5.1 Emulation Attacks

The number of responses of a memory-based PUF is limited by the number of its memory elements, which enables reading out all PUF responses and to emulate the PUF. Further, most delay-based PUFs are subject to emulation or model building attacks [116, 148, 128, 121, 160] that exploit the linear structure of existing delay-based PUFs to create mathematical models that allow estimating the PUF response to a given PUF challenge. A number of countermeasures inserting non-linearity into the PUF's delay circuit have been proposed [116, 128, 127, 129]. However, Rührmair et al. [160] show that most of these approaches are ineffective against emulation attacks based on machine learning techniques, such as logistic regression and evolution strategies. The complexity of these attacks can be increased by obfuscating the actual PUF response using cryptography [67] (cf. Section 6.6.1) or XOR networks [127].

6.5.2 Side Channel Attacks

Side channel attacks are hardware attacks that aim to extract secret data, such as cryptographic keys, from an electronic component. Hereby, the adversary observes the behavior (such as the power consumption, electromagnetic radiation and/or timing behavior) of the component while it is using the secret data to be extracted. Since the behavior of the component is typically dependent on the data processed, it can leak information on this data. The fundamental observation is that processing a data bit of value 1 typically

consumes a different amount of power and/or time than processing a data bit of value 0.

PUFs are typically used in combination with fuzzy extractors (cf. Section 6.3) and most PUF-based applications (cf. Section 6.4.2) require the PUF responses to be secret. Hence, side channel attacks against PUF-based systems typically target the fuzzy extractor to gather challenge/response pairs and other information that eases emulation attacks on the underlying PUF.

Research on the side channel analysis of PUFs and fuzzy extractors has been recently started and there are only a few published results. Karakoyunlu et al. [102] and Merli et al. [133] show side channel attacks on implementations of common fuzzy extractors. Furthermore, Merli et al. [133] discuss potential side channel leakages of various PUF types. However, all known side channel attacks on PUF-based systems target the fuzzy extractor and are independent of the underlying PUF construction.

6.5.3 Fault Injection Attacks

Fault injection attacks aim to prompt erroneous behavior in a device by manipulating it in some way and, when combined with cryptanalysis, can lead to key recovery attacks. Faults may be injected in many ways, e.g., by operating the device in extreme environmental conditions or by injecting transient faults into specific components of the device.

Attempts to operate the PUF outside its normal operating envelope, e.g., by varying its supply voltage or ambient temperature, typically affect the challenge/response behavior and thus the robustness and unpredictability property of the PUF, as our evaluation results show (cf. Section 7.4). Further, the impact of remanence decay effects on the unpredictability and robustness of memory-based PUFs [182, 170, 88, 172, 89] enables denial-of-service and fault injection attacks to recover the PUF response. We recently presented [145] a fault-injection attack based on the remanence decay in volatile memory and showed how it can be exploited effectively to launch a non-invasive cloning attack against SRAM PUFs. We validated the approach against two SRAM PUF implementations in 65 nm CMOS ASICs.

Moreover, since implementations of fuzzy extractors (cf. Section 6.3) and the underlying error correction algorithms are typically not resistant to fault injection attacks and exhibit data-dependent behavior, fault injection attacks can cause unintended leakage of PUF-related secret information, such as cryptographic keys bound to the PUF. In

particular, most fuzzy extractors are not secure in case the helper data can be modified by the adversary [27]. Thus, robust fuzzy extractors [56] should be used to prevent manipulations of the helper data.

6.6 Advanced PUF Concepts

This section discusses advanced PUF concepts that enhance the security properties and extend the functionality of standard PUFs.

6.6.1 Controlled PUFs

Most delay-based PUFs are subject to model building attacks that allow emulating the PUF in software (cf. Section 6.5.1). One approach to counter this problem are Controlled PUFs [68] that use cryptography in hardware to hide the actual PUF response from the adversary. Controlled PUFs typically apply a cryptographic hash function to the PUF challenges and/or responses, which introduces non-linearity and breaks up the link between the actual PUF response and the output of the controlled PUF. Clearly, this does not address the fundamental weakness of delay-based PUFs. Moreover, to maintain verifiability of the controlled PUF outputs, error correction must be applied before the noisy responses of the underlying PUF are processed by the cryptographic operation, which increases the complexity of the overall construction. Further, to protect against emulation attacks, the cryptographic component and the error-correction mechanism as well as their connecting wires must be protected against invasive and side channel attacks (cf. Sections 6.5.2 and 6.5.3), which may be hard to achieve in practice.

6.6.2 Reconfigurable PUFs

So far, most existing PUFs exhibit a static behavior while a variety of applications benefits from the availability of PUFs whose characteristics can be changed dynamically, i.e., reconfigured, after deployment. For instance, PUF-based key storage (cf. Section 6.4.2) and PUF-based cryptographic primitives [4] may require that previous secrets derived from the PUF cannot be retrieved any more (e.g., to achieve forward secrecy). Another example are solutions to prevent downgrading of software [114] by binding the software to a certain hardware configuration, such as a PUF, which requires the PUF behavior to be irreversibly altered upon installation of a new software update.

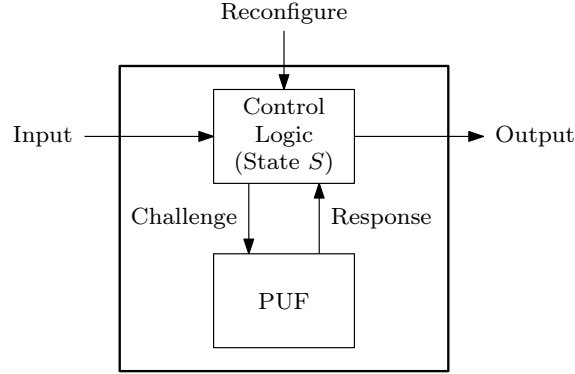


Figure 6.2: Concept of Logically Reconfigurable PUFs

Unfortunately, all known implementations of physically reconfigurable PUFs rely on optical mechanisms, reconfigurable hardware (such as FPGAs) or novel memory technologies [114], which all have several limitations in practice. In particular, optical PUFs cannot be easily integrated into integrated circuits and require expensive and error-prone evaluation equipment, while FPGA-based solutions cannot be realized with non-reconfigurable hardware (such as ASICs) that is commonly used in practice [125].

In this context, several attempts to emulate physically reconfigurable PUFs have been made. One of the first proposals was integrating a floating gate transistor into the delay lines of an Arbiter PUF, which allows physically changing the challenge/response behavior of the PUF based on some state maintained in non-volatile memory [119, 120]. Other approaches restrict access to the interface of the PUF and use part of the PUF challenge as reconfiguration data [116, 114], which, however, works only for certain PUF types.

We recently formalized the concept and the security properties of Logically Reconfigurable PUFs (LR-PUFs) [108]. In contrast to static PUFs, LR-PUFs can be dynamically reconfigured after deployment such that their challenge/response behavior changes in a random manner without replacing or physically modifying the PUF. The idea is amending a conventional PUF with a stateful control logic that transforms challenges and responses of the PUF (cf. Figure 6.2). We presented and evaluated two different constructions for LR-PUFs that are simple, efficient and can easily be implemented.

6.7 Conclusion and Open Problems

Physically Unclonable Functions (PUFs) are a very promising approach to increase the security of embedded systems, such as RFID tags. They open new directions towards lightweight, secure and privacy-preserving protocols based on physical assumptions and cost-effective tamper-evident storage for cryptographic secrets that even cannot be learned or reproduced by the manufacturer of the corresponding PUF.

PUF realizations require careful statistical testing before they can be safely deployed to real security-critical products. Even though experimental results have been reported in the literature for some PUF implementations, it is difficult to compare them due to varying test conditions and different analysis methods. We present a new evaluation methodology in Chapter 7 that allows a more precise assessment of the unpredictability property of PUF responses. Further, we provide a large-scale security analysis of ASIC implementations of the five most popular electronic PUF types that allows for the first time a fair comparison of these PUFs.

There is no widely accepted security framework for PUFs while most PUF security models in the literature are not general enough and exclude certain PUF types, do not reflect all properties of real PUF implementations or include security parameters that cannot be determined for real PUF implementations. Typically, idealized PUF models that capture the desired properties of an ideal PUF component are used in the literature. Since it is unclear whether such ideal PUFs exist, a common evaluation framework for the analysis of real PUF implementations is needed to design practical PUF-based security solutions. We present a PUF security framework providing security definitions that are compliant to standard game-based cryptographic security models and that allow for evaluating and quantifying the properties of PUF implementations in Chapter 8.

Since PUFs are bound to the device in which they are embedded, no other entity can verify the response of a PUF to a given challenge without knowing an authentic challenge/response pair (CRP) in advance, which may lead to scalability problems in practice. Current PUF-based protocols aim at circumventing this problem by providing the verifier (e.g., an RFID reader) with a database that contains a set of CRPs that act as reference values for the responses of the interrogated PUF. However, this approach opens the possibility for denial-of-service and replay-attacks. We present two scalable and lightweight PUF-based mutual authentication protocols for RFID that overcome the drawbacks of existing approaches in Chapter 9.

7 Security Evaluation of PUF Implementations on ASIC

In this chapter, we present the first large-scale security analysis of ASIC implementations of the five most popular electronic PUF types. Our analysis is based on PUF data obtained at different operating conditions from 96 ASICs housing multiple PUF instances, which have been manufactured in TSMC 65 nm CMOS technology. We present an evaluation methodology and quantify the robustness and unpredictability properties of PUF responses, which are fundamental for the integration of PUFs into cryptographic primitives and protocols, such as authentication schemes. Since all PUFs have been implemented in the same ASIC and analyzed with the same evaluation methodology, our results allow for the first time a fair comparison of their properties.

Remark. The results presented in this chapter are due to the author of this work and the result of many intensive discussions with Stefan Katzenbeisser, Ünal Kocabaş and Ahmad-Reza Sadeghi (all TU Darmstadt, Germany). The design and parts of the implementation of the evaluation framework are due to the author of this work. The ASICs and evaluation boards used in this work have been designed by our partners Intel, Intrinsic ID, KU Leuven and Sirrix AG. Part of the raw PUF data used for the evaluation has been provided by Intrinsic ID. Vladimir Rožić and Ingrid Verbauwhede (both KU Leuven, Belgium) provided detailed information on the implementations of the PUFs in the ASIC. Ünal Kocabaş set up the test environment and collected parts of the raw PUF data used for the PUF evaluation. Further, he implemented parts of the evaluation framework. Parts of this chapter have been published in [107].

7.1 Motivation and Contribution

We present the first large-scale security analysis of ASIC implementations of the five most popular electronic PUF types, including different delay-based PUFs (Arbiter and Ring Oscillator PUFs) and different memory-based PUFs (SRAM, Flip-flop and Latch PUFs). Hereby, we focus on robustness and unpredictability, which are the most vital PUF properties in many security-critical applications. The ASICs have been manufactured in

TSMC 65 nm CMOS technology within a multi-project wafer run and contain multiple implementations of the same PUF design. Our analysis is based on PUF data obtained from 96 ASICs at different temperatures, supply voltages and noise levels that correspond to the corner values typically tested for consumer-grade IT products. In this context, we developed an evaluation methodology for the empirical assessment of the robustness and unpredictability properties of PUFs. Since all PUFs have been implemented in the same ASIC and analyzed with the same methodology, our results allow for the first time a fair comparison of the robustness and unpredictability of these PUFs.

Our evaluation results show that all PUFs in the ASIC are sufficiently robust for practical applications. However, not all of them achieve the unpredictability property. In particular, the responses of Arbiter PUFs have very low entropy, while the entropy of Flip-flop and Latch PUF responses are affected by temperature variations. In contrast, the Ring Oscillator and SRAM PUFs seem to achieve all desired properties of a PUF: Their challenge/response behavior hardly changes under different operating conditions and the entropy of their responses is quite high. Furthermore, the responses generated by different Ring Oscillator and SRAM PUF instances seem to be independent, which means that the adversary cannot predict the response of a PUF based on the challenge/response pairs of another PUF. However, the min-entropy, i.e., the minimum number of random bits observed in a response of the Ring Oscillator PUF, is low, which means that some responses can be guessed with high probability.

An independent evaluation of electronic PUFs on a 65 nm ASIC has been presented at the same time as our work [19]. However, their evaluation of the unpredictability of PUF responses is based only on Hamming distances and does not consider entropy.

7.2 The PUF ASIC

Our analysis is based on data obtained from 96 ASICs that have been manufactured in TSMC 65 nm CMOS technology within a Europractice multi-project wafer run. The ASIC has been designed within the UNIQUE research project by our partners Intel, Intrinsic ID and KU Leuven. Each ASIC implements multiple instances of three different memory-based PUFs (SRAM, Flip-flop and Latch PUFs) and two different delay-based PUFs (Ring Oscillator and Arbiter PUFs). The main characteristics and the number of PUF instances in the ASICs are shown in Table 7.1. Furthermore, the ASIC is equipped with an active core that emulates the noisy working environment of a microprocessor.

Table 7.1: Physically Unclonable Functions (PUFs) Implemented in the 96 ASICs

PUF class	PUF type	Number of PUF instances per ASIC	Total number of instances	Challenge space size	Response space size
Delay-based	Arbiter	256	24,576	2^{64}	2
	Ring Oscillator	16	1,536	$32,640 \approx 2^{15}$	2
Memory-based	SRAM	4 (8 kB)	384	2^{11}	2^{32}
	Flip-flop	4 (1 kB)	384	2^8	2^{32}
	Latch	4 (1 kB)	384	2^8	2^{32}

When enabled, this core continuously performs AES encryptions.

The implementation of the Arbiter PUF follows the basic approach presented by Lee et al. [116] and consists of 64 delay elements and an arbiter. The delay elements are connected in a line, forming two delay paths with an arbiter placed at the end. Each challenge corresponds to a different configuration of the delay paths. More detailed, each delay element has two inputs and two outputs and can be configured to map inputs to outputs directly (challenge bit 0) or to switch them (challenge bit 1). During the read-out of the PUF response, the input signal propagates along both paths and, depending on which of the paths is faster, a single response bit is generated. To ensure that the delay difference results from the manufacturing process variations rather than the routing of the metal lines, a symmetric layout for the delay elements and full-custom layout blocks were used. Further, to reduce any bias, the capacitive loads of the connecting metal wires was balanced and a symmetric NAND-latch was used as arbiter.

The Ring Oscillator PUF uses the design by Suh et al. [180]. Each Ring Oscillator PUF consists of 256 ring oscillators and a control logic which compares the frequency of two different oscillators selected by the PUF challenge. Depending on which of the oscillators is faster, a single response bit is generated. The individual ring oscillators are implemented using layout macros to ensure that all oscillators have exactly the same design, which is fundamental for the correct operation of the Ring Oscillator PUF.

The memory-based PUFs are implemented as arrays of memory elements (SRAM cells, latches, flip-flops). All these memory elements are bi-stable circuits with two stable states corresponding to a logical 0 and 1. After power-up, each memory element enters either of the two states. The resulting state depends on the manufacturing process variations and the noise in the circuit. When challenged with a memory address, the PUF returns

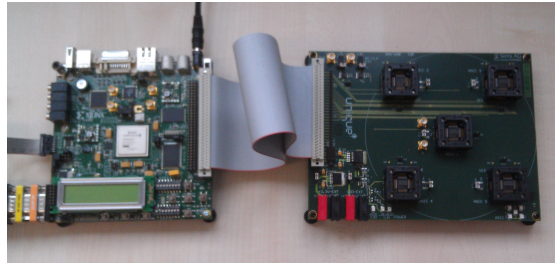


Figure 7.1: Test setup with Xilinx Virtex 5 FPGA (left) and ASIC evaluation board with five PUF ASICs (right).

the 32 bit data word at that address. The implementations of the memory-based PUFs follow the SRAM PUF design by Holcomb et al. [88], the Flip-flop PUF design by Maes et al. [124] and the Latch PUF design by Su et al. [179]. Latch and Flip-flop PUFs are implemented using the standard cells from TSMC’s 65 nm low-power library. The placement and implementation of the SRAM cells of the SRAM PUF has been done by TSMC’s memory compiler. The Latch and Flip-flop PUFs are based on standard cells using a clustered strategy where all latches or flip-flops of the same PUF instance are grouped together in single block.

The test setup consists of an ASIC evaluation board designed and manufactured by our project partner Sirrix AG, a Xilinx Virtex 5 FPGA and a workstation (cf. Figure 7.1). Each evaluation board can take up to five ASICs and allows controlling the ASIC supply voltage with an external power supply. The interaction with the evaluation board and the ASICs is performed by the FPGA, which is connected to a workstation that controls the PUF evaluation process and stores the raw PUF responses obtained from the ASICs. The tests at different temperatures have been performed in a climate chamber.

7.3 Evaluation Methodology

Many PUF-based applications require PUF responses to be reliably reproducible while at the same time being unpredictable [125, 3]. Hence, our empirical evaluation focuses on robustness and unpredictability.

Table 7.2: Robustness Test Cases

Test Case	Active Core		Ambient Temperature			Supply Voltage			Iter. k
	Off	On	-40°C	$+25^\circ\text{C}$	$+85^\circ\text{C}$	1.08 V	1.2 V	1.32 V	
E_1	×		×			×			20
E_2	×		×				×		40
E_3	×		×					×	20
E_4	×			×		×			30
E_5	×			×			×		60
E_6	×			×				×	30
E_7	×				×	×			20
E_8	×				×		×		40
E_9	×				×			×	20
E_{11}		×		×			×		60

7.3.1 Robustness Analysis

Robustness is the property that a PUF always generates responses that are similar to the responses generated during the enrolment of the PUF. Note that PUFs should fulfil this property under different operating conditions, such as different temperatures, supply voltages and noise levels. The robustness of PUFs can be quantified by the bit error rate $\text{BER} := \frac{\text{HD}(y_{E_i}, y_{E_5})}{|y_{E_5}|}$, which indicates the fraction of bits of a PUF response y_{E_i} that are different from the response y_{E_5} observed during enrolment. We determine the maximum BER of all PUF instances in all ASICs based on challenge/response pairs collected at different ambient temperatures (-40°C to $+85^\circ\text{C}$), supply voltages ($\pm 10\%$ of the nominal 1.2 V) and noise levels (active core enabled and disabled), which correspond to the corner values that are typically tested for consumer grade IT products. This shows the impact of the most common environmental factors on the BER of each PUF type. We did not test different noise levels at different temperatures and supply voltages since most PUFs (except the Arbiter PUF) turned out to be hardly affected by even the maximum amount of noise the active core can generate. An overview of all test cases considered for robustness is given in Table 7.2. We estimate the BER of all PUFs in all ASICs using the following procedure:

Step 1: Sample challenge set generation. A sample challenge set X' is generated for each PUF type (Arbiter, Ring Oscillator, SRAM, Flip-flop and Latch PUF) and used in all subsequent steps. For all but the Arbiter PUF the complete challenge space is used

as a sample set. Since the Arbiter PUF has an exponential challenge space, we tested it for 13,000 randomly chosen challenges, which is a statistically significant subset and representative for the whole challenge space.

Step 2: Enrolment. For each PUF instance, the response y_i to each challenge $x_i \in X'$ is obtained under nominal operating conditions (test case E_5) and stored as the reference value in a database DB_0 .

Step 3: Data acquisition. For all test cases E_p in Table 7.2, each PUF instance is evaluated k times on each $x_i \in X'$ and its responses are stored in a database DB_p for $p = 1, \dots, 11$.

Step 4: Analysis. For each PUF instance, the maximum BER between its responses in DB_0 and its responses in DB_1, \dots, DB_{11} over all $x_i \in X'$ is computed.

7.3.2 Unpredictability Analysis

Unpredictability ensures that the adversary cannot efficiently compute the response of a PUF to an unknown challenge, even if he can adaptively obtain a certain number of other challenge/response pairs from the same and other PUF instances [3]. This is important in most PUF-based applications, such as authentication protocols, where the adversary can forge the authentication when he can predict a PUF response. Note that unpredictability should be independent of the operating conditions of the PUF, which could be exploited by an adversary.

The unpredictability of a PUF implementation can be estimated empirically by applying statistical tests to its responses and/or based on the complexity of the best known attack against the PUF [125, 3]. Statistical tests, such as the DIEHARD [131] or NIST [162] test suite, can in principle be used to assess the unpredictability of PUF responses. However, since these test suites are typically based on a series of stochastic tests, they can only indicate whether the PUF responses are random or not. Moreover, they require more input data than the memory-based PUFs and Ring Oscillator PUFs in the ASIC provide.

Similar as in symmetric cryptography, the unpredictability of a PUF can be estimated based on the complexity of the best known attack. There are attacks (cf. Section 6.5.1)

Table 7.3: Unpredictability Test Cases

Test Case	Active Core		Ambient Temperature			Supply Voltage		
	Off	On	−40 °C	+25 °C	+85 °C	1.08 V	1.2 V	1.32 V
E_{13}	×		×				×	
E_{14}	×			×			×	
E_{15}	×				×		×	
E_{16}	×			×		×		
E_{17}	×			×				×

against delay-based PUFs that emulate the PUF in software and allow predicting PUF responses to arbitrary challenges. These attacks are based on machine learning techniques that exploit statistical deviations and/or dependencies of PUF responses. However, emulation attacks have been shown only for simulated PUF data and it is currently unknown how these attacks perform against real PUFs [160].

Another approach is estimating the entropy of the PUF responses based on experimental data. In particular *min-entropy* indicates how many bits of a PUF response are uniformly random. The entropy of PUFs can be approximated using the context-tree weighting (CTW) method [198], which is a data compression algorithm that allows assessing the redundancy of bit strings [92, 186, 79, 188].

We assess the unpredictability of PUFs using Shannon entropy, which is a common metric in cryptography and allows establishing relations to other publications that quantify the unpredictability of PUFs using entropy [187, 180, 88, 3]. We estimate the entropy and min-entropy of the responses of all available PUFs. Specifically, we first check whether PUF responses are biased by computing their Hamming weight and estimate an upper bound of the entropy of PUF responses using a compression test. Eventually, we approximate the entropy and min-entropy of the responses of all available PUFs. Our entropy estimation is more precise than previous approaches since it considers dependencies between the individual bits of the PUF responses. Furthermore, to get an indication of whether responses of *different* PUF instances are independent, we compute the Hamming distance between the responses of different PUF instances.

We assess the unpredictability of all available PUFs at different temperatures and supply voltage levels (cf. Table 7.3) to determine the effects of environmental variations on the unpredictability using the following procedure:

Step 1: Sample challenge set generation. For each PUF type, a sample challenge set X' is generated that is used in all subsequent steps. For all but the Arbiter PUF, the complete challenge space is used as a sample challenge set. Since the Arbiter PUF has an exponential challenge space, we again test it only for 13,000 challenges. The subsequent analysis steps require $X' := \{x' \in X'' \mid \text{HD}(x, x') \leq k\}$, which includes a set X'' of randomly chosen challenges and all challenges that differ in at most k bits from the challenges in X'' (which may be known to the adversary).

Step 2: Data acquisition. For all test cases E_q in Table 7.3, each PUF instance is evaluated on each $x_i \in X'$ and the responses y are stored in a database DB_q .

Step 3: Analysis. For each test case E_q , the responses in DB_q are analyzed as follows:

Step 3a: Hamming weight. For each PUF instance, the average Hamming weight of all its responses y_i in DB_q is computed, which indicates whether the responses are biased towards 0 or 1.

Step 3b: CTW compression. For each PUF instance, a binary file containing all its responses in DB_q is generated and compressed using the context-tree weighting (CTW) algorithm [197]. The resulting compression rate is an estimate of the upper bound of the entropy of the PUF responses.

Step 3c: Entropy estimation. For each PUF instance, the entropy and min-entropy of all its responses in DB_q is estimated as detailed in the next paragraph.

Step 3d: Hamming distance. For each PUF type, the Hamming distance $\text{HD}(y, y')$ of all pairs of responses (y, y') in DB_q generated by pairwise different PUF instances for the same challenge x is computed. While all previous steps consider only responses of the *same* PUF instance, the Hamming distances indicate whether responses of *different* PUF instances are independent. This is important to prevent the adversary from predicting the responses of one PUF implementation based on the challenge/response pairs of another (e.g., his own) PUF implementation, which would contradict the unpredictability property.

Entropy Estimation. Let x be the PUF challenge for which the adversary should predict the response y . Further, let $Y(x)$ be the random variable representing y . Moreover, let $W(x)$ be the random variable representing the set of all responses of the PUF except y , i.e., $W(x) = \{y' | y' \leftarrow \text{PUF}(x'); x' \in X \setminus \{x\}\}$. We are interested in the conditional entropy

$$\mathbf{H}(Y|W) = - \sum_{x \in X} \Pr[Y(x), W(x)] \cdot \log_2 \Pr[Y(x)|W(x)] \quad (7.1)$$

and the conditional min-entropy

$$\mathbf{H}_\infty(Y|W) = -\log_2 \left(\max_{x \in X} \{ \Pr[Y(x)|W(x)] \} \right), \quad (7.2)$$

which quantify the average and minimal number of bits of y , respectively, that cannot be predicted by the adversary, even in case all other responses in $W(x)$ are known. Hence, $2^{-\mathbf{H}_\infty(Y|W)}$ is an information-theoretic upper bound for the probability that an adversary guesses the PUF response y to the challenge x .

However, computing Equations 7.1 and 7.2 for $W(x)$ is difficult since (1) the sizes of the underlying probability distributions are exponential in the response space size and (2) the complexity of computing $\mathbf{H}(Y|W)$ grows exponentially with the challenge space size of the PUF to be analyzed. Hence, Equations 7.1 and 7.2 can at most be estimated by making assumptions on the physical properties of the PUFs that reduce the size of $W(x)$. In the following, we explain how we estimated these entropies for each PUF type and discuss the underlying assumptions.

Assumptions on memory-based PUFs. A common assumption on memory-based PUFs is that spatially distant memory cells are independent [125, 3]. A similar assumption has been used by Holcomb et al. [88], who estimate the entropy of SRAM PUF responses based on the assumption that individual bytes of SRAM are independent. However, physically neighboring memory cells can strongly influence each other, in particular when they are physically connected.¹ Hence, our entropy estimation considers dependencies between neighboring memory cells (which could be exploited by an adversary) while assuming that spatially distant memory cells are independent. More specifically, we compute the entropy of the PUF response bit $Y_{i,j}$ of the memory cell at

¹SRAM cells are typically arranged in a matrix where all cells in a row are connected by a word line and all cells in a column are connected by a bit line.

row i and column j of the underlying memory under the worst case assumption that the values of all neighboring memory cells $W'(x) = (Y_{i-1,j}, Y_{i,j+1}, Y_{i+1,j}, Y_{i,j-1})$ are known, i.e., we compute Equations 7.1 and 7.2 for $W'(x)$.

Note that the bit-pattern read from an SRAM may not correspond to its physical layout. This means that neighboring bits in the bit-pattern read from the SRAM may be stored in physically distant SRAM cells. Hence, before estimating the entropy using the described approach, we had to reorder the bit-patterns read from the SRAM to match the physical layout of the SRAM cells.

Assumptions on Ring Pscillator PUFs. The Ring Oscillator PUFs in the ASICs compare the oscillation frequency of two ring oscillators O_i and O_j selected by the PUF challenge $x = (i, j)$ and return a response $Y(i, j)$, depending on which of the two oscillators was faster. Since neighboring ring oscillators may affect each other (e.g., by electromagnetic induction), we consider the potential dependency between the frequencies of neighboring oscillators and assume that the frequency of spatially distant oscillators is independent. Thus, we compute Equations 7.1 and 7.2 for $W'(i, j) = (Y_{i-2,j}, Y_{i-1,j}, Y_{i+1,j}, Y_{i+2,j})$.

Assumptions on Arbiter PUFs. Arbiter PUFs measure the delay difference of two delay lines that are configured by the PUF challenge. The individual delays caused by the switches and their connections are additive, which implies that the PUF response y to a challenge x can be computed if a sufficient number of responses to challenges that are close to x are known. Hence, we compute Equations 7.1 and 7.2 for $W'(x) = \{y' \leftarrow \text{PUF}(x') \mid x' \in X', \text{HD}(x, x') \leq k\}$, which corresponds to the worst case where the adversary knows responses to challenges that differ in at most k bits from the challenge which of the response he must guess. Specifically, in our evaluation we use X consisting of 200 randomly chosen challenges and $k = 1$.

Computing the entropy. To compute the entropy and min-entropy (Equations 7.1 and 7.2) for each test case E_q , we first estimate $\Pr[x = Y(x), w = W(x)]$ for each $x \in X'$ by dividing the number of observations of each tuple (x, w) in database DB_q by the size of the sample challenge set X' . Further, to compute $\Pr[x = Y(x) \mid w = W(x)] = \Pr[x = Y(x), w = W(x)] / \Pr[w = W(x)]$, we estimated $\Pr[w = W(x)]$ by dividing the number of observations of each tuple $(Y(x), w = W(x))$ in database DB_i by the size of X' .

Eventually, we used these empirically estimated probability distributions to compute the entropy and min-entropy according to Equations 7.1 and 7.2, respectively.

7.4 Evaluation Results

We applied the evaluation methodology described in Section 7.3 to all PUF instances in all ASICs. Most of our results are illustrated using *bean plots* [101] that allow an intuitive visualization of empirical probability distributions (cf. Figures 7.2 to 7.5). Each bean shows two distributions, smoothed by a Gaussian kernel to give the impression of a continuous distribution, together with their means indicated by black bars. The distribution in black on the left side typically corresponds to data collected under normal PUF operating conditions, while the one in gray on the right side corresponds to some other test case in Table 7.2 or 7.3. This allows an easy visualization of the PUF behavior under changing environmental conditions. Each plot contains several beans that correspond to the different PUF types available in the ASICs, which allows an easy comparison of the results for different PUF types.

7.4.1 Robustness Results

We computed the bit error rate (BER) under varying environmental conditions (cf. Table 7.3). Our results show that all Arbiter, Ring Oscillator and SRAM PUF instances have a very similar BER, while there is a big variability in the BERs of the Flip-flop and Latch PUF instances (cf. Figure 7.2). Further, the BER of the Arbiter, Ring Oscillator and SRAM PUF instances is below 10% for all test cases, which can be handled by common error correction schemes, such as fuzzy extractors (cf. Section 6.3). The BER of most PUFs depends on the operating temperature. Compared to $+25^{\circ}\text{C}$ (test case E_5), at -40°C (test case E_2) the BER of the Flip-flop and Latch PUF increases significantly, while the BER of the Ring Oscillator and SRAM PUF increases only slightly and the BER of the Arbiter PUF hardly changes (cf. Figure 7.2a). A similar behavior of the BERs can be observed at $+85^{\circ}\text{C}$ (test case E_8 , Figure 7.2b). All PUFs in all ASICs turned out to be robust against variations of their supply voltages. Compared to nominal operating conditions (test case E_5), the distributions of the BERs only slightly increase when varying the supply voltage by 10% (test cases E_4 and E_6 , Figure 7.2c). The Arbiter PUF exhibits a significantly increased BER when operated in a noisy working environment (test case E_{11} , Figure 7.2d) while there is no significant change of the BER of all

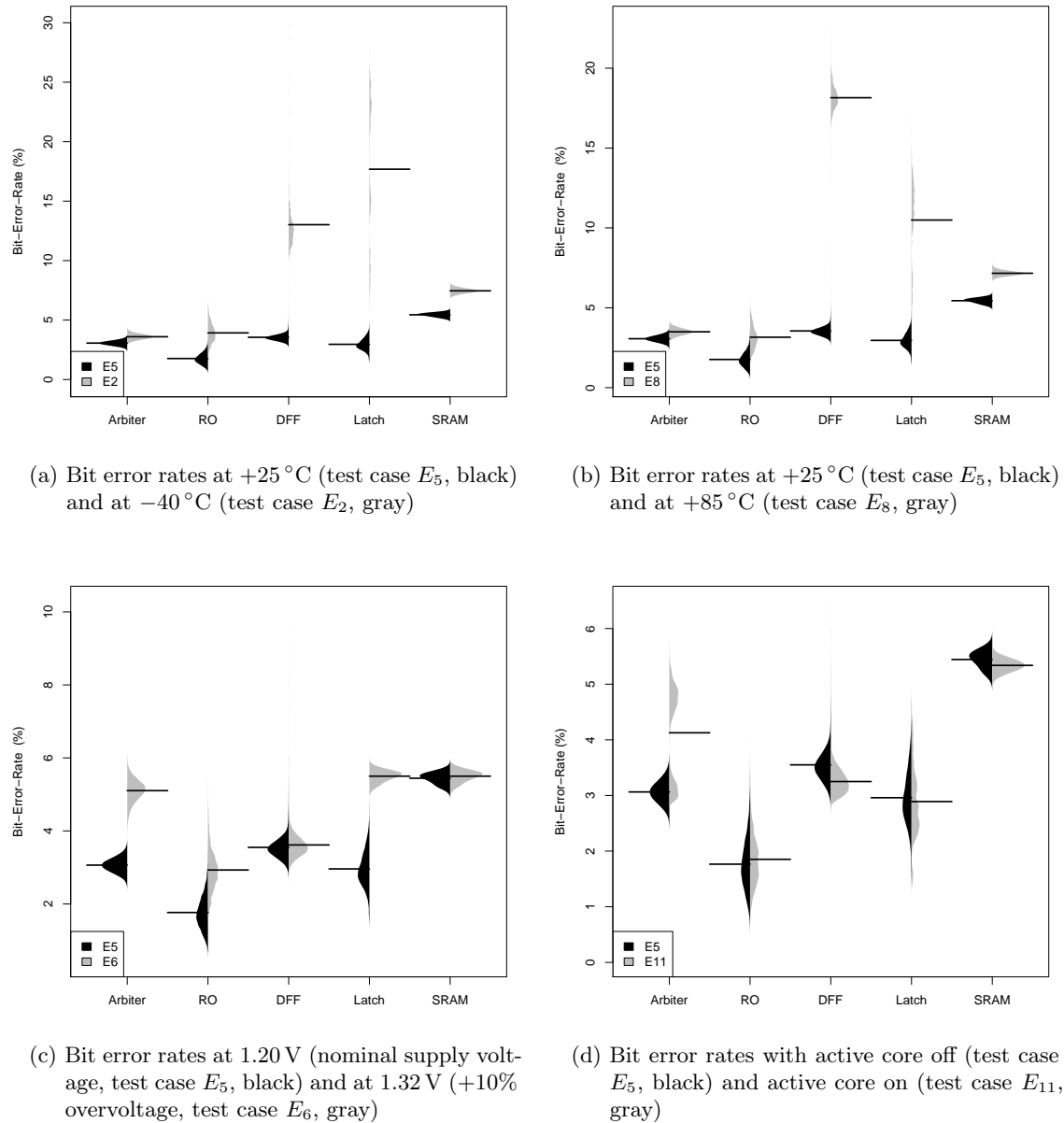


Figure 7.2: Distribution of the bit error rates (BER) in percent over all PUF instances at different ambient temperatures, supply voltages and noise levels. The two peaks of the BER distribution of the Arbiter PUF in Figure 7.2d show that those Arbiter PUFs that are spatially close to the active core are more affected by noise than those farther away.

Table 7.4: CTW Compression Test Results

Test Case	Size of the PUF response after the CTW compression in percent				
	Arbiter PUF	RO PUF	Flip-Flop PUF	Latch PUF	SRAM PUF
E_{13}	—	0.77	0.77	0.84	1.00
E_{14}	0.51	0.77	0.87	0.70	1.00
E_{15}	—	0.77	0.98	0.53	1.00
E_{16}	0.53	0.77	0.88	0.69	1.00
E_{17}	0.49	0.77	0.87	0.71	1.00

other PUFs. Hereby, we observed that the BER of those Arbiter PUF instances that are spatially close to the active core significantly changes, while those that are farther away are not directly affected.

7.4.2 Unpredictability Results

In this section, we present the results of our unpredictability analysis. Due to the time-limited access to the climate chamber, the data required to analyze the unpredictability of the Arbiter PUF at -40°C and at $+85^{\circ}\text{C}$ is not available. However, we show the results for normal operating conditions and different supply voltages.

Hamming weights. To get a first indication of randomness in the PUFs, we computed the Hamming weight of their responses as described in Section 7.3.2. Our results show that Ring Oscillator and SRAM PUF responses are close to the ideal Hamming weight of 0.5, independent of the operating conditions (cf. Figure 7.3), which indicates that their responses may be random. The Hamming weight of the Flip-flop PUF and Latch PUF responses strongly depends on the ambient temperature (cf. Figures 7.3a and 7.3b) and is clearly biased. Supply voltage variations (test cases E_{16} and E_{17}) have no significant impact on the Hamming weight of the responses of any of the PUF instances in the ASIC (cf. Figures 7.3c and 7.3d).

CTW compression. The context-tree weighting (CTW) compression test gives a good indication of the upper bound of the entropy of PUF responses. The higher the compression rate, the lower the entropy of the PUF. The results of this test are shown in Table 7.4 and confirm the Hamming weight test results: The compression rate of the Ring Oscillator and SRAM PUF responses is invariant for all test cases; the compres-

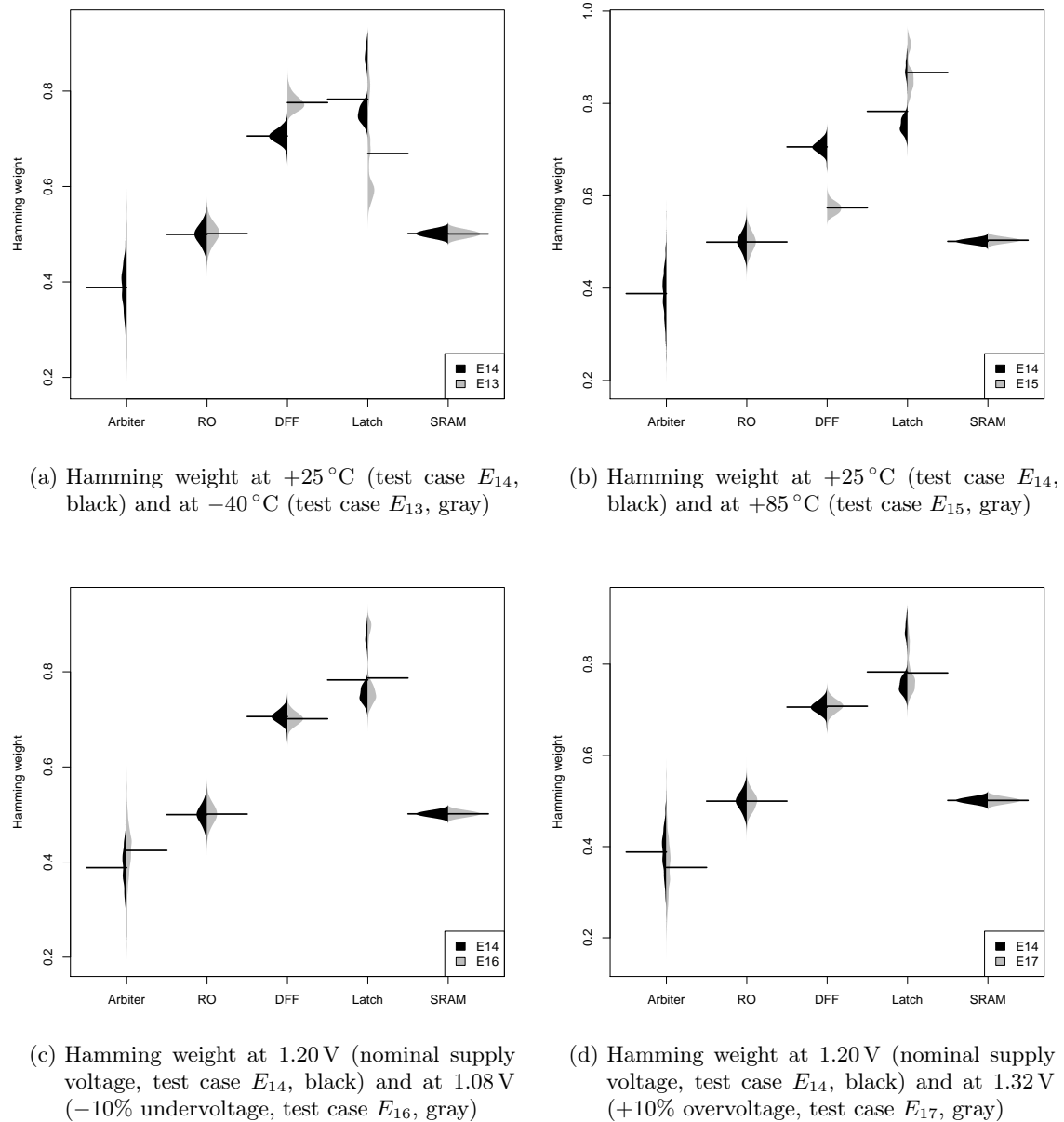
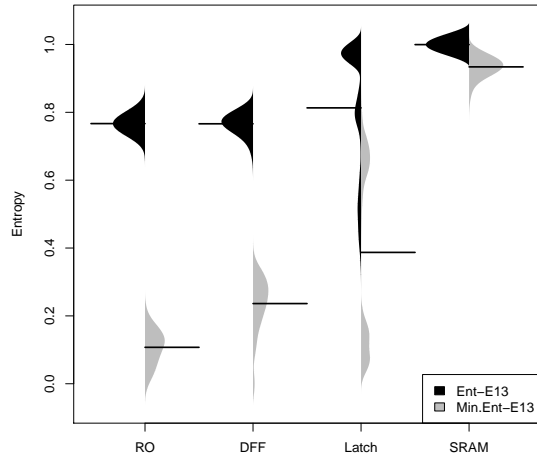


Figure 7.3: Distribution of the Hamming weight over all PUF instances at different ambient temperatures and supply voltages. The two peaks of the Hamming weight distribution of the Latch PUF may come from the fact that one of the four Latch PUF instances on each ASIC is implemented in a separate power domain.

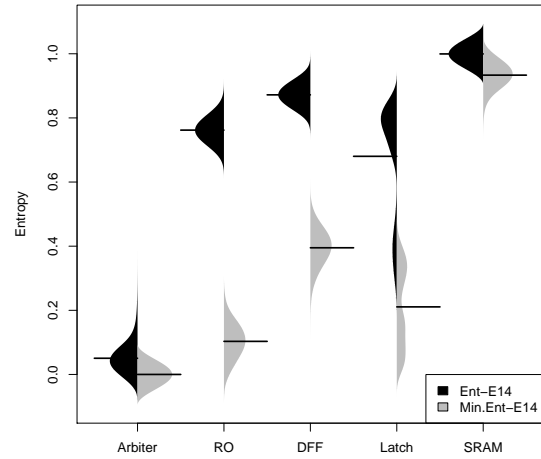
sion rates of the Flip-flop and Latch PUF responses do not change for different supply voltages (test cases E_{16} and E_{17}) but vary with the ambient temperature (test cases E_{13} , E_{14} and E_{15}). The compression rate of the SRAM PUF responses strongly indicates that these responses are uniformly random, while there seem to be some dependencies in the responses generated by all other PUFs.

Entropy estimation. The results of the entropy estimation described in Section 7.3.2 confirm the results of all previous tests and provide more insights into the entropy and min-entropy of the PUF responses (cf. Figure 7.4). The entropy of responses corresponding to neighboring Arbiter PUF challenges is remarkably low, which confirms the high prediction rate of the emulation attacks on Arbiter PUFs reported in the literature [160]. The entropy and min-entropy of the Ring Oscillator and SRAM PUF responses is invariant to temperature (test cases E_{13} , E_{14} and E_{15} , Figures 7.4a to 7.4c) and supply voltage (test cases E_{16} and E_{17} , Figure 7.4d) variations. Moreover, the entropy and min-entropy of Flip-flop and Latch PUFs vary with the operating temperature (test cases E_{13} , E_{14} and E_{15} , Figures 7.4a to 7.4c) and are constant for different supply voltages (test cases E_{16} and E_{17} , Figure 7.4d).

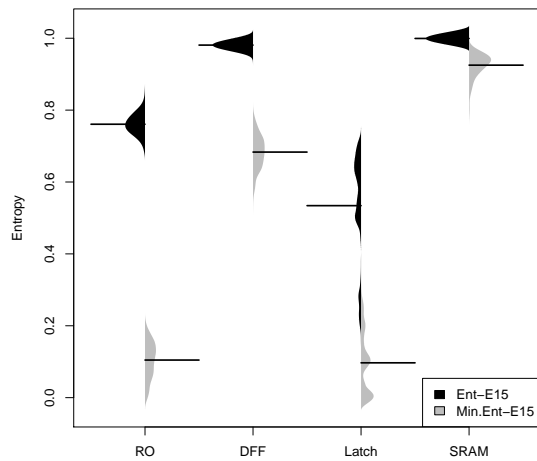
Hamming distances. The Hamming distance test (cf. Section 7.3.2) gives an indication of whether the responses generated by different PUF instances to the same challenge are independent. Our results show that, independent of the ambient temperature (test cases E_{13} , E_{14} and E_{15}) and supply voltage (test cases E_{16} and E_{17}), the responses of different Ring Oscillator and SRAM PUF instances have the ideal Hamming distance of 0.5, while there seem to be dependencies between the responses generated by different Arbiter PUF instances to the same challenge (cf. Figure 7.5). The Hamming distance of the responses of the Flip-flop PUFs changes for different temperatures and supply voltages. At $+85^\circ\text{C}$ (test case E_{15} , Figure 7.5b) the Hamming distance of the Flip-flop PUF is ideal, while it is biased towards zero at -40°C (test case E_{13} , Figure 7.5a). Moreover, at 1.08 V (-10% undervoltage, test case E_{16} , Figure 7.5c) we observed a bias of the Hamming distance towards one, while the Hamming distance at 1.32 V ($+10\%$ overvoltage, test case E_{17} , Figure 7.5d) is similar to the distribution at nominal operating conditions (test case E_{14}). The Hamming distance of the responses of the Latch PUFs are biased towards zero and invariant for different supply voltages.



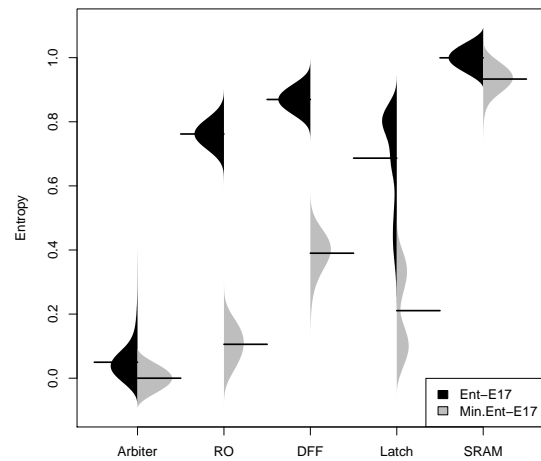
(a) Entropy (black) and min-entropy (gray) at -40 °C (test case E_{13})



(b) Entropy (black) and min-entropy (gray) at +25 °C (test case E_{14})

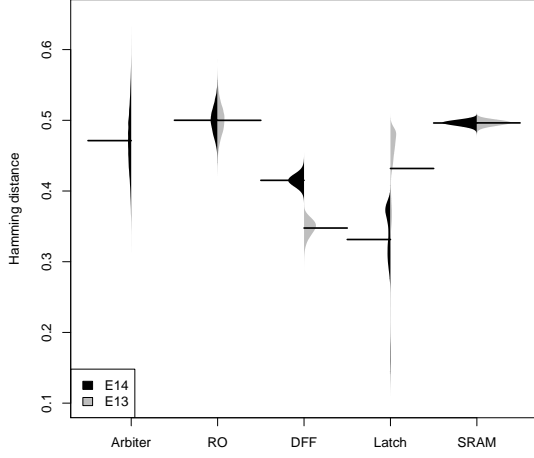


(c) Entropy (black) and min-entropy (gray) at +85 °C (test case E_{15})

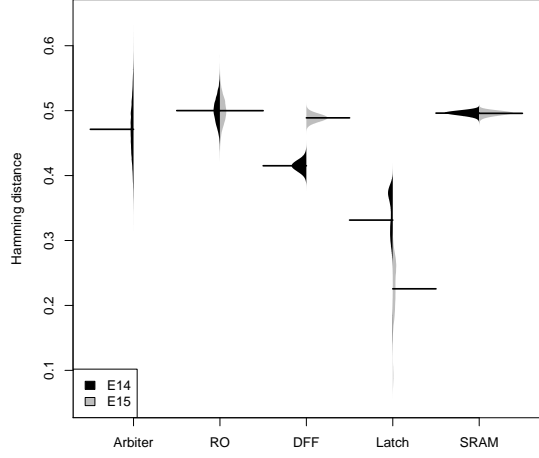


(d) Entropy (black) and min-entropy (gray) at 1.32 V (+10% overvoltage, test case E_{17})

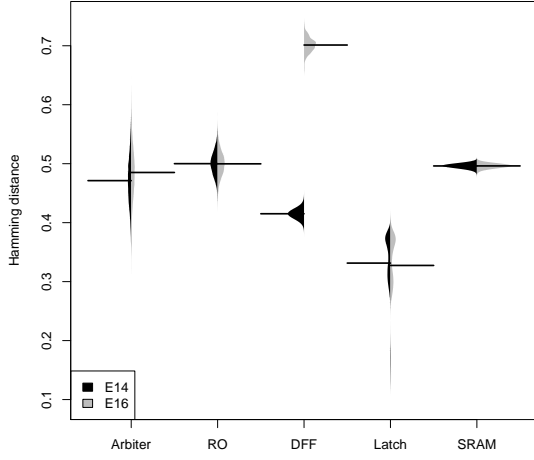
Figure 7.4: Distribution of the entropy (black) and min-entropy (gray) over all PUF instances at different ambient temperatures and supply voltages. Note that the parts of the graphs showing an entropy/min-entropy < 0 and > 1 are drawing errors due to the Gaussian kernel used to smooth the discrete distributions to give the impression of a continuous distribution.



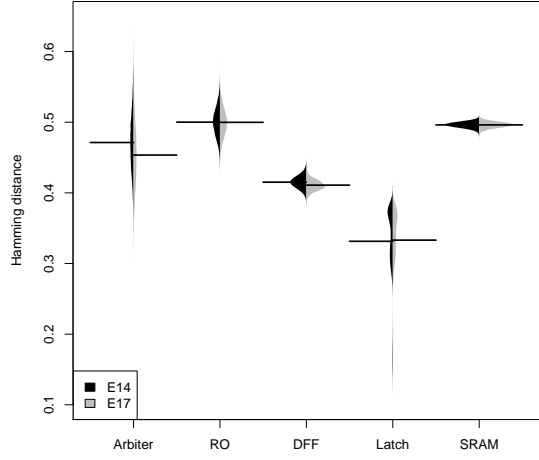
(a) Hamming distance at $+25^{\circ}\text{C}$ (test case E_{14} , black) and at -40°C (test case E_{13} , gray)



(b) Hamming distance at $+25^{\circ}\text{C}$ (test case E_{14} , black) and at $+85^{\circ}\text{C}$ (test case E_{15} , gray)



(c) Hamming distance at 1.20 V (nominal supply voltage, test case E_{14} , black) and at 1.08 V (-10% undervoltage, test case E_{16} , gray)



(d) Hamming distance at 1.20 V (nominal supply voltage, test case E_{14} , black) and at 1.32 V ($+10\%$ overvoltage, test case E_{17} , gray)

Figure 7.5: Distribution of the Hamming distance over all PUF instances at different ambient temperatures and supply voltages.

7.4.3 Discussion

Our results show that Arbiter, Ring Oscillator and SRAM PUFs are more robust to temperature variations than the Latch and Flip-flop PUFs. This could be due to the dual nature of these PUFs, i.e., the two delay paths, two ring oscillators and the symmetrical structure of the SRAM cells, respectively. As discussed in Section 7.2, we do not have access to the internal circuit diagrams and layout of the standard cells provided by TSMC and thus can only speculate about the transistor schematics of the flip-flops and latches. Standard cell libraries typically use implementations based on transmission gates, which are more compact than static latches or flip-flops with a dual structure and there is no duality or symmetry in these transistor schematics. Further, the results of the Hamming weight and Hamming distance tests indicate that the unpredictability of PUFs with a dual structure are less affected by temperature variations.

The entropy of the Arbiter PUF is remarkably low, which can be explained by the linear structure of this PUF. Note that in the Arbiter PUF implementation, two signals travel along two delay paths and finally arrive at an arbiter (cf. Section 7.2). In case the delay difference δ_t of the two paths is greater than the setup time t_{setup} plus the hold time t_{hold} of the arbiter, the PUF response will be correctly generated according to which signal arrives first. However, in case $\delta_t < t_{\text{setup}} + t_{\text{hold}}$, the arbiter will be in the metastable state and the PUF response will depend on the bias of the arbiter caused by manufacturing process and/or layout variations of the arbiter and the noise in the circuit. A limited number of simulations (with 20 PUFs for 3 challenges) including extracted post layout parasitics were performed before the tape-out of the ASIC to estimate this effect.

Since the Arbiter PUF design is based on delay accumulation, it is very susceptible to emulation attacks (cf. Section 6.5.1). An example illustrating this fact is the case where two challenges differ in only the last bit. In this case, signals will travel along the same paths through 63 delay elements and only in the last element the paths will be different. If the adversary knows the outcome for one challenge, he can guess the outcome of the other one with high probability, which might explain the low entropy and min-entropy of the Arbiter PUFs.

7.4.4 Summary

The Arbiter PUF responses have a very low entropy and their use in applications with strict unclonability and unpredictability requirements should be carefully considered. Fur-

ther, the Arbiter PUFs are susceptible to changes of their supply voltage and to environmental noise, which significantly increases the bit error rate of the PUF. However, the bit error rate stays within acceptable bounds and can be compensated by existing error correction mechanisms.

The Flip-flop and Latch PUFs are susceptible to temperature variations, which have a significant effect on the bit error rate and the unpredictability of the PUF responses. Hence, Flip-flop and Latch PUFs should not be used in an environment where the adversary can lower the ambient temperature of the PUF, reducing the entropy of the PUF responses.

The SRAM and Ring Oscillator PUFs achieve almost all desired properties of a PUF: The bit error rate does not change significantly under different operating conditions, the entropy of the PUF responses is high and the responses generated by different PUF instances seem to be independent. However, the Ring Oscillator PUF exhibits a low min-entropy, which might be problematic in some applications.

7.5 Conclusion

We performed the first large-scale analysis of the five most popular PUF types (Arbiter, Ring Oscillator, SRAM, Flip-flop and Latch PUFs) implemented in ASIC. Our analysis is based on PUF data obtained from 96 ASICs, each housing several PUF instances. Our results allow for the first time a fair comparison of these PUFs. In this context, we presented an evaluation methodology for the empirical assessment of the robustness and unpredictability properties of PUFs that are fundamental in most applications of PUFs.

Our results show that the SRAM and Ring Oscillator PUFs seem to achieve all desired properties of a PUF. However, the Arbiter PUFs have a very low entropy and the entropy of the Flip-flop and Latch PUFs is susceptible to temperature variations. Hence, the suitability of these PUFs for security-critical applications, such as authentication or key generation must be carefully considered.

8 Formal Model of Physically Unclonable Functions

We present a formal foundation for security primitives based on PUFs, focussing on the main properties at the heart of most published works on PUFs: robustness, unclonability and unpredictability. This work allows for a meaningful security analysis of security primitives taking advantage of physical properties, becoming increasingly important in the development of the next generation of secure information systems. So far, our framework has been used to estimate the robustness and unclonability properties of image-based PUFs [174], the design of anti-counterfeiting mechanisms [173] and physical hash functions [58].

Remark. The PUF security framework presented in this section is the shared result of an intense research collaboration between the author of this work, Frederik Armknecht (University of Mannheim, Germany), Roel Maes (KU Leuven, Belgium), Ahmad-Reza Sadeghi (TU Darmstadt, Germany), François-Xavier Standaert (Université Catholique de Louvain, Belgium). Parts of this chapter have been published in [3].

8.1 Motivation and Contribution

Currently only rudimentary security models for PUFs exist, limiting the confidence in the security claims of PUF-based security primitives (cf. Section 6.1 and the overview of PUF models in [3]). A useful model should at the same time (1) define the security properties of PUFs abstractly and naturally, allowing to design and formally analyze PUF-based security solutions and (2) provide practical quantification tools allowing to evaluate PUF instantiations.

Exploiting physical properties in security systems raises important formalization problems. The core issues are to determine which properties of physical objects need to be defined and to find efficient ways to guarantee them in practice. In other words, one of the main challenges for using PUFs in future security applications is to properly integrate them into complex systems, where some of their physical properties can be a real advantage compared to purely algorithmic solutions. In this respect, useful and reasonable

security definitions of PUFs should be both (1) sound for cryptographers, in order to allow the analysis of PUF-based cryptographic systems and (2) empirically verifiable, such that the security levels guaranteed by the physics can be evaluated (or at least be lower bounded). These challenges give a strong motivation for introducing a security model for PUFs that unifies previous formalization attempts and at the same time satisfies both requirements. For this purpose, our rationale is based on the following observations:

1. It is difficult to argue about the physical properties of an object, e.g., compared to classical cryptography, where explicit security parameters can do an excellent job in this respect.
2. It is unknown if the properties expected for PUFs, such as unpredictability or unclonability, relate to any exponentially hard problem. While this situation can be unsatisfying from a theoretical point of view, it is in fact similar to the situation of many primitives used in applied cryptography. For example, there is no exponential hardness problem on which current block ciphers are based, e.g., the AES is only *expected* to provide a security level of roughly 2^{128} operations.
3. The interface of PUFs to the outside world usually does not directly access the physics but uses some mathematical post-processing of the PUF outputs (which we denote as extractor algorithm).

As a consequence of (1) and (2), our focus is to start with a set of three basic properties allowing the design of hybrid systems combining PUFs with classical algorithms and to formalize PUFs by security notions similar to those of, e.g., block ciphers, with constant security levels that can be properly quantified in a physical counterpart to cryptanalysis. First, PUFs must be robust, i.e., able to provide stable outputs, since non-robust PUFs would significantly harm the efficiency of the underlying system. Robustness essentially captures the resilience of a PUF system to noisy measurements. Next, we investigate formal definitions of unclonability, which is a central property of PUFs that cannot be guaranteed by purely algorithmic solutions. Having improved arguments of unclonability, quantified within a sound model, would better motivate the use of PUFs in many security applications. Third, we propose a definition of unpredictability of PUF outputs, which is the weakest cryptographic property that could be expected from PUFs. While unpredictability could also be guaranteed by algorithmic means, we believe that the inherent physical randomness provided by PUFs is worth to be exploited as well. As a consequence

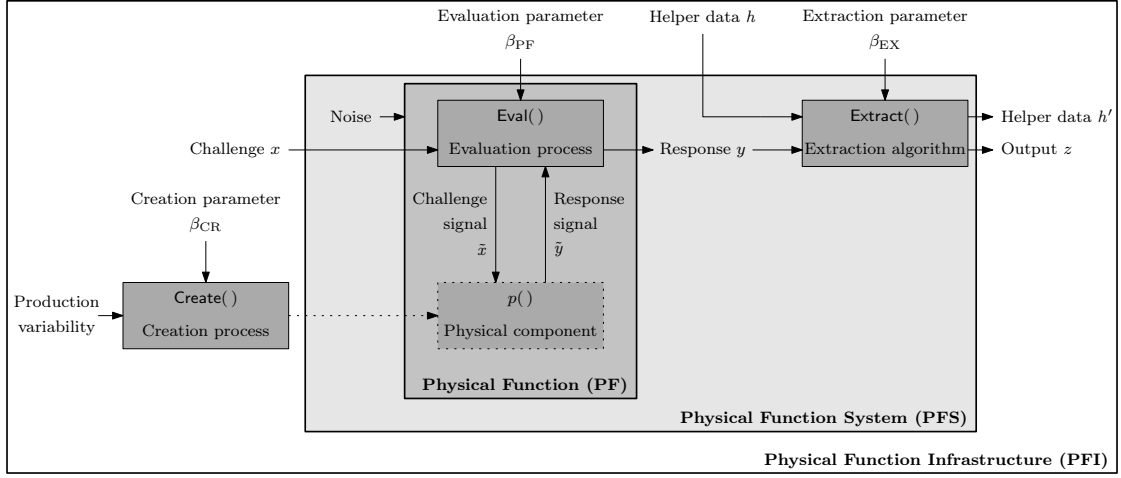


Figure 8.1: Generic Framework for Physical Functions

of (3), we finally propose to define these cryptographic properties as a function of the extractor algorithm instead of a plain PUF.

8.2 Framework for Physical Functions

8.2.1 Background and Rationale

In this section, we explain the components and procedures relevant for deploying physical functions. Observe that we focus not only on PUFs but on physical functions in general where unclonability is only one possible security property. Before we provide formal definitions, we give an overview of our framework, which is depicted in Figure 8.1 that shows all components necessary for creating, evaluating and post-processing the output of a physical function. In the following, we explain each of these components separately.

Physical Function

A *Physical Function* (PF) consists of a *physical component* p that can be stimulated with some *challenge signal* \tilde{x} which makes p respond with a corresponding *response signal* \tilde{y} . In addition to the physical component p , a PF contains an *evaluation procedure* Eval that, on input a digital representation x of \tilde{x} , stimulates the physical component with \tilde{x} and obtains the resulting response signal \tilde{y} . Finally, Eval returns a digital representation y of \tilde{y} . Note that with *procedure* we denote a probabilistic polynomial time

algorithm that may involve some physical process such as the evaluation of a PUF. The challenge/response behavior of a PF heavily relies on the properties of the physical component p , uncontrollable random noise (e.g., thermal noise and measurement uncertainties) and an evaluation parameter β_{PF} (e.g., a quantization factor) chosen by the PF manufacturer. Observe that the same physical component p can yield completely different PFs if combined with different evaluation procedures. This fact should be representable by a comprehensive model.

Extraction Algorithm

Although the notion of a physical function suggests differently, a PF is not a function in the classical sense. The main difference is that, when challenged with the same challenge x twice, a PF may produce different responses y . This is because the challenge/response behavior of a PF heavily relies on the physical properties of its physical component p , which is subject to uncontrollable random noise. The effects of noise can be removed up to a certain threshold by an *extraction algorithm* **Extract** (e.g., a fuzzy extractor [57, 56]), which maps slightly different responses y to the same challenge x to a unique output z according to some *extraction parameter* β_{EX} , which is typically chosen by the PF manufacturer or the PF user (i.e., the entity that integrates the PUF into a higher-level protocol or algorithm). We assume that the extraction parameter specifies both the deployed extraction algorithm and all possible parameters (e.g., the number of output bits) of the chosen **Extract** algorithm. The **Extract** algorithm can be executed in two different modes: *setup* and *reconstruction*. If a challenge x is requested for the first time, the setup mode is used to generate an output z and some appropriate *helper data* h' . Later, when the challenge x is requested again together with the helper data $h = h'$, the reconstruction mode is used to recreate z . The purpose of the helper data h' is to twofold [57]: (1) h' supports the extraction algorithm **Extract** in recreating the same output z for a challenge x and (2) h' allows to bind given values (e.g., cryptographic keys) to a PUF.

Physical Function System

As explained above, a PF is usually coupled with an appropriate extraction algorithm. Indeed, in a typical application scenario, a user will be only aware of the challenges given to the PF and the output returned by the extraction algorithm. Furthermore, for

almost all relevant security notions, both the deployed PF and the extraction algorithm determine whether a security property is given or not. Therefore, it is a natural choice to abstract away the physical function PF and the extraction algorithm **Extract** and consider their combination as one single building block. We term this a *Physical Function System* (PF system). Consequently, we will mostly refer to PF systems only and refer to the underlying PF or extraction algorithm only if necessary.

Creation Process

The creation of the physical component p of a physical function PF is the result of a *creation process* **Create**, usually performed by the manufacturer of PF. The result of this process depends on a creation parameter β_{CR} that is chosen by the PF manufacturer and some uncontrollable production variability.

Physical Function Infrastructure

We call the combination of all previously described components a *Physical Function Infrastructure* (PFI). We stress that within a PFI the creation, evaluation and extraction parameters are *fixed*. Furthermore, we assume that these parameters uniquely specify the deployed procedures, e.g., β_{PF} defines the full details of the **Eval** procedure.

8.2.2 Formalization

We now formally define the components and procedures within a physical function infrastructure as explained in Section 8.2.1.

Definition 8.1 (Physical Function). *A physical function PF is a probabilistic procedure (i.e., a probabilistic polynomial time algorithm that may involve some physical process)*

$$\text{PF}_{p,\beta_{\text{PF}}} : X \rightarrow Y,$$

where X denotes the set of challenges and Y the set of responses. Internally, a PF is the combination of a physical component p and an evaluation procedure **Eval** which takes as input an extraction parameter β_{PF} and a challenge $x \in X$, i.e.,

$$y \leftarrow \text{PF}_{p,\beta_{\text{PF}}}(x) = \text{Eval}_p(\beta_{\text{PF}}, x).$$

Usually, the specification of p and β_{PF} will be discarded in our notation, that is we simply write PF instead of $\text{PF}_{p,\beta_{\text{PF}}}$.

Definition 8.2 (Physical Function System). *A physical function system PFS is a probabilistic procedure*

$$\text{PFS}_{p,\beta_{\text{PF}},\beta_{\text{EX}}} : X \times (H \cup \{\emptyset\}) \rightarrow Z \times H,$$

where X is the set of challenges, H the set of helper data values, \emptyset the empty string and Z the set of outputs. Internally, a PF system is the combination of a physical function $\text{PF} = \text{PF}_{p,\beta_{\text{PF}}}$ (Definition 8.1) and an extraction algorithm Extract which is determined by an extraction parameter β_{EX} :

$$(z, h') \leftarrow \text{PFS}_{p,\beta_{\text{PF}},\beta_{\text{EX}}}(x, h) = \text{Extract}_{\beta_{\text{EX}}}(\text{PF}_{p,\beta_{\text{PF}}}(x), h).$$

Hereby, we require that if $h \neq \emptyset$, then $h' = h$. Only in case $h = \emptyset$, a new helper data h' is generated for x . In the following we omit the internal components and abbreviate $\text{PFS} = \text{PFS}_{p,\beta_{\text{PF}},\beta_{\text{EX}}}$.

Note that $h = \emptyset$ means that Extract should be executed in setup mode and generate a new helper data h with regard to challenge x . In case $h \neq \emptyset$, Extract should be executed in reconstruction mode and recreate output z associated with the challenge x and the helper data h . Note that, for the sake of consistent notation, in this case we require $h' = h$ to be returned by Extract .

Definition 8.3 (Creation Process). *A creation process Create is a probabilistic procedure that, on input of a creation parameter β_{CR} , produces a physical component p (Definition 8.1).*

Definition 8.4 (Physical Function Infrastructure). *A physical function infrastructure \mathcal{F} refers to a fixed creation process Create (Definition 8.3) and the set of all PF systems PFS (Definition 8.2) where the physical component p is the result of Create, i.e.,*

$$\mathcal{F}_{\beta_{\text{CR}}} = (\text{Create}, \{\text{PFS}_{p,\beta_{\text{PF}},\beta_{\text{EX}}} : p \leftarrow \text{Create}(\beta_{\text{CR}})\}),$$

where β_{CR} , β_{PF} and β_{EX} are fixed.

8.3 Robustness

8.3.1 Rationale

As explained in Section 8.2, a PF might respond to the same challenge with different responses when queried several times. However, if these responses are “similar”, it is possible to overcome this problem by using an appropriate extraction algorithm. By robustness, we refer to the property that former outputs of a PF system can be reconstructed at a later time. Obviously, a certain level of robustness is a necessary prerequisite for using PF systems as functions in the classical sense.

Robustness could refer to at least two properties: (1) the ability to reconstruct the output of a PF system that has been produced by the setup mode or (2) the ability to always recreate the same output in reconstruction mode (which may be different from the output in setup mode). We decided for the first option for two reasons: First, one can show that a high probability for (1) implies also a high probability for (2). Furthermore, (1) directly reflects the basic criterion that is necessary in a typical PUF-based key generation scenario.

8.3.2 Formalization

Following the consideration mentioned above, we formally define the robustness of a PF system as follows:

Definition 8.5 (Robustness). *Let PFS be a PF system (Definition 8.2) and let $x \in X$ be a challenge. The challenge robustness of PFS w.r.t. x is defined as the probability*

$$\rho_{\text{PFS}}(x) := \Pr [z' = z | (z, h) \leftarrow \text{PFS}(x, \emptyset) \wedge (z', h') \leftarrow \text{PFS}(x, h)] .$$

This means that robustness is the probability that an output generated by **Extract** in reconstruction mode matches the output generated earlier by **Extract** in setup mode.

In practice, the best estimate of the challenge robustness is the sample mean over many evaluations of the same challenge on the same PF system. For cases where it is important that each challenge has at least a certain robustness, the notion of *minimum robustness* is introduced:

Definition 8.6 (Minimum Robustness of a PF System). *The minimum robustness of a*

PF system PFS (Definition 8.2) w.r.t. to a set of challenges $X' \subseteq X$ is defined as

$$\rho_{\text{PFS}}^{\min} := \min \{ \rho_{\text{PFS}}(x) : x \in X' \}.$$

In some cases it may be difficult to estimate the minimum robustness. Actually, from a practical point of view, it can be sufficient that the average challenge robustness over many challenges of a PF system is high enough. This is where the notion of *average robustness* comes in:

Definition 8.7 (Average Robustness of a PF System). *The average robustness of a PF system (Definition 8.2) w.r.t. a set of challenges $X' \subseteq X$ is defined as*

$$\rho_{\text{PFS}}^{\text{avg}} := \sum_{x \in X'} \Pr[x \xrightarrow{\$} X'] \cdot \rho_{\text{PFS}}(x).$$

So far we considered PF systems where the underlying physical function PF is fixed. Moreover, it is important to consider the probability of finding PF systems with a certain minimum/average robustness within a given PF infrastructure. The corresponding terminology is given in the following definitions:

Definition 8.8 (Minimum Robustness of a PF Infrastructure). *Consider a PF infrastructure $\mathcal{F} = (\text{Create}, P)$ (Definition 8.4), where $P = \{\text{PFS}_{p, \beta_{\text{PF}}, \beta_{\text{EX}}} : p \leftarrow \text{Create}(\beta_{\text{CR}})\}$. The minimum robustness of \mathcal{F} is defined as*

$$\rho_{\mathcal{F}}^{\min} := \min \{ \rho_{\text{PFS}}^{\min} : \text{PFS} \in P \}.$$

Analogously, we define the average robustness of a PF infrastructure as follows:

Definition 8.9 (Average Robustness of a PF Infrastructure). *Let $\mathcal{F} = (\text{Create}, P)$ be a PF infrastructure (Definition 8.4), where $P = \{\text{PFS}_{p, \beta_{\text{PF}}, \beta_{\text{EX}}} : p \leftarrow \text{Create}(\beta_{\text{CR}})\}$. The average robustness of \mathcal{F} is defined as*

$$\rho_{\mathcal{F}}^{\text{avg}} := \sum_{\text{PFS} \in P} \Pr[\text{PFS} \xrightarrow{\$} P] \cdot \rho_{\text{PFS}}^{\text{avg}}.$$

Here, $\text{PFS} \xrightarrow{\$} P$ denotes the event that a random physical component p has been created, i.e., $p \leftarrow \text{Create}(\beta_{\text{CR}})$ and that a PF system PFS has been generated based on p , i.e., $\text{PFS} = \text{PFS}_{p, \beta_{\text{PF}}, \beta_{\text{EX}}}$ for a fixed β_{PF} and β_{EX} .

8.4 Physical Unclonability

8.4.1 Rationale

As this work is motivated by the increasing usage of *Physically Unclonable Functions*, it is a natural choice to include unclonability into the model, which is the key property of PUFs that cannot be achieved by algorithmic solutions. In this section, we formally define the notion of *physical* unclonability. We stress that we consider only clones on the physical level and exclude mathematical clones. This restriction is motivated by the fact that an adversary in general has different possibilities for creating (i.e., cloning) a PF system that shows the “same” behavior as another PF system. For instance, the adversary could choose an **Extract** algorithm that maps all inputs to the same output. Clearly, two different PF systems using this **Extract** algorithm would behave exactly the same, independent of the underlying PFs. It is obvious that protection against such attacks can only be provided by mechanisms outside of the PF system. In general, while physical unclonability is an intrinsic feature, this is not true for mathematical unclonability, which hence is outside of the scope of a PF security model. We propose a definition of physical unclonability that can informally be stated as follows: *A PF system PFS' is a physical clone of another PF system PFS if both PF systems show the same behavior and deploy the same **Extract** algorithm.* By the second condition, we guarantee that we consider clonability on a physical level only.

It remains to discuss how to formalize the notion of “same behavior”. Recall that PFs are assumed to be noisy in general, which raises the question of when two PFs can be considered being the same. A good starting point is to consider at first only one PF system. Recall that the extraction procedure is deployed to make a PF system “as deterministic as possible”. Nonetheless, in certain cases, the same PF system might produce the same output twice only with a certain probability. We referred to this probability as the robustness of the PF system and termed it $\rho_{PFS}(x)$ in dependence of the considered challenge x (cf. Definition 8.5). Intuitively, a clone PFS' cannot be more similar to the corresponding original PF system PFS than PFS itself. On the other hand, any PF system should be formally seen as a clone of itself. Therefore, the robustness marks a natural upper bound on “how similar a clone can become” and it seems to be natural to integrate the notion of robustness into the definition of clones.

Another aspect that needs to be considered is the following: Depending on the use

case, only the responses of PFS to a subset of challenges might be known at all. Thus, any other PF system PFS' that coincides on this subset of challenges could be seen as a clone. Therefore, it is sufficient that the definition of a clone captures only the set of challenges $X' \subseteq X$ that are relevant with regard to the underlying use case.

Note that a cloning attack can have different meanings:

- *Selective cloning* refers to the event that for a *given* PF system PFS a clone PFS' is constructed.
- *Existential cloning* means that two arbitrary PF systems PFS and PFS' are produced, where one is the clone of the other.

The difference between selective and existential cloning is that in the latter case no “original PF system” is given and instead, the adversary is free to choose which PF system is cloned. Observe that this classification has some similarities to the security properties established for digital signatures and message authentication codes (MACs).

8.4.2 Formalization

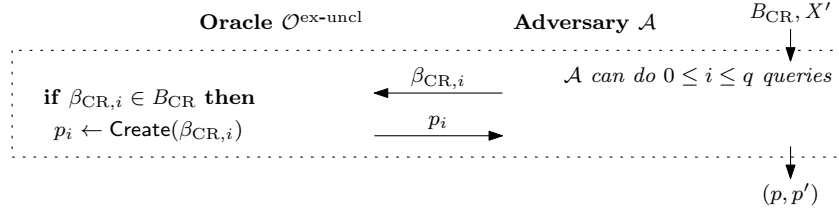
We start with formalizing the notion of a clone:

Definition 8.10 (Physical Clone). *Let β_{PF} and β_{EX} be a fixed evaluation and extraction parameter, respectively. Moreover, let $\text{PFS} = \text{PFS}_{p, \beta_{\text{PF}}, \beta_{\text{EX}}}$ and $\text{PFS}' = \text{PFS}_{p', \beta_{\text{PF}}, \beta_{\text{EX}}}$ be two PF systems (Definition 8.2) that are identical except of their physical component, i.e., $p \neq p'$. We define that PFS' is a δ -clone of PFS with regard to $X' \subseteq X$ if for all $x \in X'$ it holds that*

$$\Pr [z' = z | (z, h) \leftarrow \text{PFS}(x, \emptyset) \wedge (z', h') \leftarrow \text{PFS}'(x, h)] \geq \delta \cdot \rho_{\text{PFS}}(x).$$

For simplicity, we write $\text{PFS}' \stackrel{\delta, X'}{\equiv} \text{PFS}$ if this equation holds.

Next, we formalize both notions of unclonability by means of two security experiments that specify the capabilities and the goal of the adversary \mathcal{A} . On a high level, \mathcal{A} is capable of creating arbitrary physical components, which in turn determine PF systems. In practice, \mathcal{A} will be limited to a certain set of creation processes, e.g., by increasing the sensitivity of his production facility. We capture this formally by allowing \mathcal{A} to choose the creation parameter β_{CR} from a set B_{CR} of possible creation parameters. In practice B_{CR}


 Figure 8.2: Existential Unclonability Security Experiment $\mathbf{Exp}_{\mathcal{A}}^{\text{ex-uncl}}(q)$

is expected to be small. We start by defining existential unclonability, where \mathcal{A} must produce two *arbitrary* clones. In this scenario, which is depicted in Figure 8.2, \mathcal{A} can query the **Create** process for $\beta_{\text{CR}} \in B_{\text{CR}}$ to create physical components p (cf. Definition 8.3).

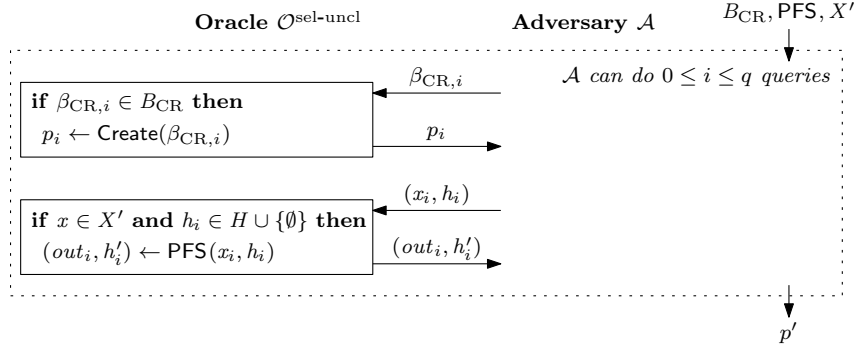
Note that a physical function p implicitly defines a PF system $\text{PFS} = \text{PFS}_{p, \beta_{\text{PF}}, \beta_{\text{EX}}}$ for some fixed evaluation and extraction parameter β_{PF} and β_{CR} , respectively (cf. Definition 8.2). Typically, only adversaries for which the time and computational effort are bounded are relevant for practice. Hence, we assume that \mathcal{A} can do at most $q \geq 2$ queries to **Create**.

Definition 8.11 (Existential Physical Unclonability). *Let B_{CR} be a set of creation parameters and let β_{PF} and β_{EX} be fixed parameters for the evaluation and extraction procedures, respectively. Note that this implicitly defines a family $\mathcal{F}_{B_{\text{CR}}} := \{\mathcal{F}_{\beta_{\text{CR}}} : \beta_{\text{CR}} \in B_{\text{CR}}\}$ of PF infrastructures (Definition 8.4). A family of PF infrastructures $\mathcal{F}_{B_{\text{CR}}}$ is called (γ, δ, q) -cloning-resistant with regard to $X' \subseteq X$, if*

$$\begin{aligned} \Pr [\text{PFS}'_{p', \beta_{\text{PF}}, \beta_{\text{EX}}} \stackrel{\delta, X'}{\equiv} \text{PFS}_{p, \beta_{\text{PF}}, \beta_{\text{EX}}} | (p, p') \leftarrow \mathbf{Exp}_{\mathcal{A}}^{\text{ex-uncl}}(q) \\ \wedge p \in [\text{Create}(\beta_{\text{CR}})] \wedge \beta_{\text{CR}} \in B_{\text{CR}} \\ \wedge p' \in [\text{Create}(\beta'_{\text{CR}})] \wedge \beta'_{\text{CR}} \in B_{\text{CR}}] \leq \gamma. \end{aligned}$$

This means that the probability that \mathcal{A} generated, as output of the security experiment depicted in Figure 8.2, two physical components p and p' which (1) are clones on the PF system level and (2) that have been created using the creation parameters $\beta_{\text{CR}} \in B_{\text{CR}}$, is less than γ . Note that Definition 8.11 covers different situations:

- *Honest manufacturer:* This case reflects the probability that an honest manufacturer creates two clones by coincidence and captures clonable PFs. In the case of $B_{\text{CR}} = \{\beta_{\text{CR}}\}$, i.e., where only one creation parameter is involved, the set $\mathcal{F}_{B_{\text{CR}}}$


 Figure 8.3: Selective Unclonability Security Experiment $\mathbf{Exp}_A^{\text{sel-uncl}}(q)$

“collapses” to a single PF infrastructure $\mathcal{F}_{\beta_{\text{CR}}}$. Likewise, \mathcal{A} can perform **Create** only with this specific creation parameter. In other words, \mathcal{A} is restricted to actions that an honest manufacturer could do within $\mathcal{F}_{\beta_{\text{CR}}}$.

- *Malicious manufacturer:* This case covers the scenario, where B_{CR} contains more than one possible choice for the creation parameter β_{CR} , which allows \mathcal{A} to influence the **Create** process in order to create a clone.

Finally, we formalize selective physical unclonability in terms of the security experiment depicted in Figure 8.3. The difference to the security experiment of existential unclonability is that \mathcal{A} is *given* a PF system **PFS** for which \mathcal{A} must create a clone. Therefore, in addition to queries to **Create**, \mathcal{A} is allowed to query **PFS** with challenges $x \in X'$. Again, we consider only restricted adversaries \mathcal{A} that can do at most $q \geq 1$ queries to **Create** and **PFS**.

Definition 8.12 (Selective Physical Unclonability). *Let B_{CR} be a set of creation parameters and let β_{PF} and β_{EX} be fixed parameters for the evaluation and extraction procedures, respectively. Moreover, let $\mathcal{F}_{B_{\text{CR}}} := \{\mathcal{F}_{\beta_{\text{CR}}} : \beta_{\text{CR}} \in B_{\text{CR}}\}$ be the corresponding set of PF infrastructures (Definition 8.4). Further, let **PFS** be a PF system (Definition 8.2) within the family of PF infrastructures $\mathcal{F}_{B_{\text{CR}}}$, i.e., $\text{PFS} \in [\text{Create}(\beta_{\text{CR}})]$ for some $\beta_{\text{CR}} \in B_{\text{CR}}$. We denote with \mathcal{A} the adversary. **PFS** is called (γ, δ, q) -cloning-resistant with regard to $X' \in X$, if*

$$\Pr [\text{PFS}'_{p', \beta_{\text{PF}}, \beta_{\text{EX}}} \stackrel{\delta, X'}{\equiv} \text{PFS}_{p, \beta_{\text{PF}}, \beta_{\text{EX}}} | p' \leftarrow \mathbf{Exp}_A^{\text{ex-uncl}}(q) \wedge p' \in [\text{Create}(\beta_{\text{CR}})] \wedge \beta_{\text{CR}} \in B_{\text{CR}}] \leq \gamma.$$

8.5 Unpredictability

8.5.1 Rationale

One common application of PUFs is to use them to securely generate secret values (e.g., cryptographic keys). Examples include secure key storage [195, 120, 184] (cf. Section 6.4.2) and hardware-entangled cryptography [4]. Such applications implicitly require that the adversary cannot predict the output of a PF system. Moreover, for typical PUF-based challenge/response identification protocols (cf. Section 6.4.1) it is important that the adversary cannot predict the response to a new challenge from previously observed challenge/response pairs. Therefore, the notion of *unpredictability* is an important property that needs to be included into a model for physical functions.

Classically, the notion of unpredictability of a random function f is formalized by the following security experiment consisting of a *learning* and a *challenge* phase. In the learning phase, \mathcal{A} learns the evaluations of f on a set of inputs $\{x_1, \dots, x_n\}$ which may be given from outside or chosen by \mathcal{A} . Then, in the challenge phase, \mathcal{A} must return $(x, f(x))$ for some $x \notin \{x_1, \dots, x_n\}$. Given that this formalization is common and widely accepted in cryptography, one may be tempted to adapt it in a straightforward manner to PUFs. This would mean to take the same definition but to consider PF systems instead of functions. However, this approach does not always make sense. First, the output of a PF system depends on a challenge x *and* some helper data h . Thus, h must be taken into account. Moreover, we stress that different applications may require different variants of unpredictability. For instance, the concept of PUF-based secure key storage (cf. Section 6.4.2) is to use a PF system for securely storing a cryptographic secret k . This secret k is usually derived from the output z of a PF system for some input x . In some cases, x is public and/or possibly fixed for all instantiations. Note that in such a scenario it is required that each device generates a different secret k for the same challenge x . Hence, the outputs of different devices (i.e., their PF systems) should be independent. This requirement is captured by the following security experiment. Given the outputs $\text{PFS}_1(x, \emptyset), \dots, \text{PFS}_n(x, \emptyset)$ of a set of PF systems to a fixed challenge x within the learning phase, the adversary \mathcal{A} has to predict the output $\text{PFS}(x, \emptyset)$ for another PF system $\text{PFS} \notin \{\text{PFS}_1, \dots, \text{PFS}_n\}$ in the challenge phase. Clearly, there is a fundamental difference between the classical definition of unpredictability and this security experiment: In the original definition of unpredictability, \mathcal{A} is given the evaluation of *one* PF system

on *many* challenges, while in the latter experiment \mathcal{A} learns the evaluation of *many* PF systems on *one* fixed challenge.

Obviously, a useful definition of unpredictability of a PF system should cover both unpredictability in the original sense *and* independence of the outputs of different PF systems. Therefore, we define a security experiment that involves the following sets: (1) let P_L be the set of PF systems that are allowed to be queried by \mathcal{A} in the learning phase; (2) let P_C be the set of PF systems that are allowed to be queried by \mathcal{A} in the challenge phase; and (3) let X be the set of challenges that are allowed to be queried by \mathcal{A} during the whole experiment. Now we consider two extreme cases:¹

1. *Independence of the outputs of a single PF system:* Consider the case, where $P_L = P_C = \{\text{PFS}\}$ consists of one single PF system only, while X contains several challenges. During the learning phase, the adversary \mathcal{A} learns $\text{PFS}(x_i)$ for several challenges $x_i \in X$. Later, in the challenge phase, \mathcal{A} has to predict $\text{PFS}(x)$ for a new challenge $x \in X$. It is easy to see that this is the direct translation of the classical unpredictability experiment described at the beginning of this section to the scenario of physical function systems.
2. *Independence of the outputs of different PF systems:* Now consider the scenario, where $X = \{x\}$ consists of one single challenge only, while P_L and P_C contain several PF systems. In this case, during the learning phase, \mathcal{A} learns $\text{PFS}_i(x)$ for several different PF systems $\text{PFS}_i \in P_L$. Afterwards, in the challenge phase, \mathcal{A} has to predict $\text{PFS}(x)$ for a *new* PF system $\text{PFS} \in P_C$ that has not been queried before. This reflects the requirements of PUF-based secure key storage (cf. Section 6.4.2).

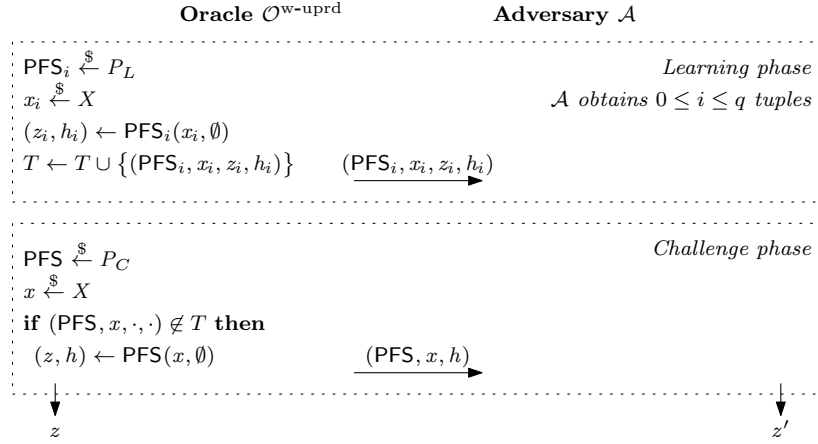
The definition of unpredictability should cover both extreme and all intermediate cases.

8.5.2 Formalization

We now define unpredictability. The definition is based on the security experiment $\text{Exp}_{\mathcal{A}}^{\text{w-uprd}}$ shown in Figure 8.4.

Definition 8.13 (Weak Unpredictability). *Let $P_L, P_C \subseteq P$ be subsets of the set of all possible PF systems. Let $T = \{ \}$ and $q \in \mathbb{N}$ with $q \geq 0$. With \mathcal{A} we denote the adversary*

¹For the sake of readability we omit the helper data here.

Figure 8.4: Weak Unpredictability Security Experiment $\mathbf{Exp}_{\mathcal{A}}^{\text{w-uprd}}(q)$

that takes part in the security experiment depicted in Figure 8.4. A PF system is weak (λ, q) -unpredictable if

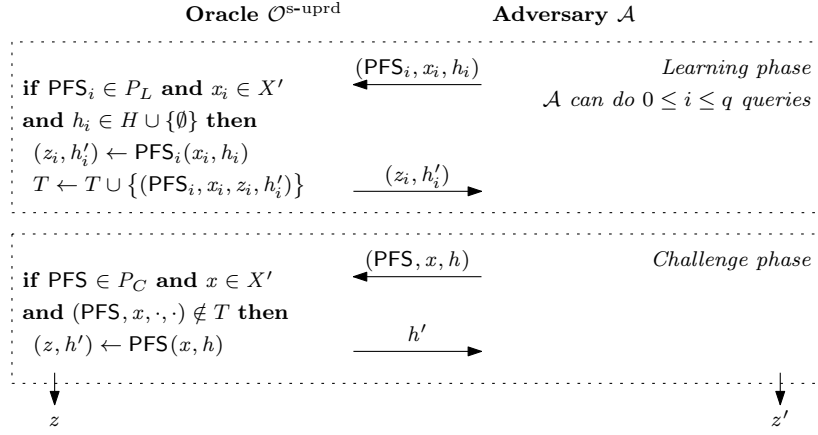
$$\Pr \left[z = z' \mid (z, z') \leftarrow \mathbf{Exp}_{\mathcal{A}}^{\text{w-uprd}}(q) \right] \leq \lambda \cdot \rho_p(x).$$

Note that the robustness of a PF system PFS is an upper bound for the predictability of the outputs of PFS. For instance, a true random number generator is a PF system with very low reliability and thus, its outputs are highly unpredictable.

While stronger notions of unpredictability exist (see below), the consideration of weak unpredictability is nonetheless important for at least the following reasons: (1) weak unpredictability is an established property in cryptography and has been used for stronger constructions [155] and (2) PF constructions may be weakly unpredictable only, e.g., Arbiter PUFs (cf. Section 7.4.2) and hence should be covered by the model.

Some use cases require a stronger notion of unpredictability where the adversary is allowed to adaptively query the PF system in the challenge phase. We therefore define strong unpredictability based on the security experiment $\mathbf{Exp}_{\mathcal{A}}^{\text{s-uprd}}$ depicted in Figure 8.5.

Definition 8.14 (Strong Unpredictability). *Let P_L be the set of PF systems that are allowed to be queried by \mathcal{A} in the learning phase and let P_C be the set of PF systems that are allowed to be queried by \mathcal{A} in the challenge phase. Moreover, let $T = \{ \}$ and $q \in \mathbb{N}$ with $q \geq 0$. With \mathcal{A} we denote the adversary that takes part in the security experiment*


 Figure 8.5: Strong Unpredictability Security Experiment $\mathbf{Exp}_{\mathcal{A}}^{\text{s-uprd}}(q)$

depicted in Figure 8.5. A PF system is strong (λ, q) -unpredictable if

$$\Pr \left[z = z' \mid (z, z') \leftarrow \mathbf{Exp}_{\mathcal{A}}^{\text{s-uprd}}(q) \right] \leq \lambda \cdot \rho_p(x).$$

8.6 Conclusion

In view of the very different physical features PUFs are based on, PUFs have mainly been developed and analyzed in independent models and under different assumptions that are specialized for the corresponding applications. This absence of a unifying view typically makes the integration of PUFs into secure information systems a difficult task, hence limiting their further development and deployment. We consequently formalized the security features of physical functions in accordance to the existing literature on PUFs. More precisely, we proposed a new general security model for physical functions, that modularly captures the most important properties required for the integration of PUFs into cryptographic primitives and security applications. Our current model focuses on the minimum requirements on PUFs and can be easily extended by defining additional security-relevant properties required by future use cases.

In fact, the extension of the model to other security properties is one of the important remaining challenges, e.g., for covering tamper-evidence, meaning the property that unauthorized manipulations of PUFs are detectable. Another challenge is to develop new cryptographic mechanisms based on PUFs where the security can be reduced to the (alleged) properties of the deployed PUFs. Our framework has been used to estimate the

robustness and unclonability properties of SRAM PUFs [3] and image-based PUFs [174] as well as in the context of designing anti-counterfeiting mechanisms [173] and physical hash functions [58].

9 PUF-enhanced Security and Privacy for RFID

We present two scalable PUF-based authentication schemes that overcome the practical and security problems of existing approaches. In contrast to existing PUF-based authentication schemes, the first protocol supports PUF-based mutual authentication between tags and readers, is resistant to emulation attacks against the underlying PUF and highly scalable since it does not require the reader to store a large number of PUF challenge/response pairs. The scheme is based on reverse fuzzy extractors [85], a new approach to correct noise in PUF responses that allows for extremely lightweight implementations on the tag. Further, it supports updating the tag authentication secrets bound to the PUF by using Logically Reconfigurable PUFs (LR-PUFs) [108, 109] that allows for changing the PUF-behaviour after deployment of the tag. The second protocol uses the PUF as a secure key storage and addresses an open question on the feasibility of destructive privacy [190], i.e., the privacy of tags that are destroyed during tag corruption.

Remark. Parts of this chapter have been published in [85] and in [168, 167].

9.1 Motivation and Contribution

The widespread use of RFID systems (e.g., for electronic transit tickets or access control) makes them attractive targets for different kinds of attacks. The most prominent example are attacks on widely used MiFare Classic tags by NXP Semiconductors [141] that allow copying (cloning) and maliciously changing the data stored on the tags [43]. Other MiFare products are claimed not to be affected. Existing solutions typically use cost-efficient tags without expensive hardware protection mechanisms [141]. Hence, the authentication secrets of these tags can often be recovered by basic side channel and invasive attacks and used to emulate the tag in software, which allows forging the information of the tag (e.g., the debit of an electronic transit ticket). To prevent such attacks, the secrets and the information of the tag should be cryptographically bound to the underlying

RFID chip such that any attempt to extract or change them permanently deactivates the tag.

In this context, Physically Unclonable Functions (PUFs) [151, 125, 3] promise to provide an effective and cost-efficient security mechanism. The common approach to authenticate a PUF-enabled tag is querying its PUF with a challenge from a pre-recorded database of PUF challenges and responses (cf. Section 6.4.1). The tag is accepted only if its response matches a PUF response in the database. However, this approach has several limitations in practice: (1) there is no support for mutual authentication between the tag and the reader; (2) most PUF types are vulnerable to emulation attacks [160] (cf. Section 6.5.1) and would allow emulating the tag in software; (3) some schemes are subject to denial-of-service attacks that permanently prevent tags from authenticating to the reader [24]; and (4) existing PUF-based authentication schemes are not scalable and allow only for a limited number of authentication protocol-runs since they rely on a database containing a large number of challenge/response pairs of the PUF of each tag. An alternative approach is using the PUF to generate the authentication secret of the tag for use in a classical authentication protocol (cf. Section 6.4.2).

In this chapter, we present two PUF-based authentication schemes for RFID: (1) a scalable and lightweight PUF-based mutual authentication protocol that overcomes the limitations of existing approaches and (2) a PUF-based authentication protocol that addresses Vaudenay’s open question on the feasibility of *destructive privacy*, i.e., privacy of tags that are destroyed during corruption.

9.2 Lightweight PUF-based RFID Authentication

We first present a new PUF-based authentication scheme that overcomes the drawbacks of existing approaches. It supports (1) PUF-based mutual authentication between tags and readers, (2) is resistant to emulation attacks and (3) supports an unlimited number of authentications without requiring the reader to store a large number of PUF challenge/response pairs. Note that the goal of this protocol is to enable lightweight and scalable PUF-based authentication of tags while the privacy of tags and users is not considered in this section.

Our protocol uses *reverse fuzzy extractors* [85], a novel approach to eliminate noise in PUF responses that moves the computationally expensive error correction process from the resource-constrained PUF-enabled tag to the more powerful RFID reader. The

resources required to implement our authentication scheme on the tag are minimal since it is based on a reverse fuzzy extractor that requires significantly less hardware resources than the error correcting mechanisms used in existing PUF-based authentication schemes or PUF-based key storage.

Remark. The PUF-based authentication scheme presented in this section is the shared result of an intense research collaboration between the author of this work, Anthony van Herrewege, Roel Maes, Roel Peeters, Ingrid Verbauwhede (all KU Leuven, Belgium), Stefan Katzenbeisser and Ahmad-Reza Sadeghi (both TU Darmstadt, Germany). The security analysis of the reverse fuzzy extractor and the design and analysis of the PUF-based authentication protocol is due to the author of this work. The key idea of the reverse fuzzy extractor and the prototype implementation of the protocol is due to Roel Maes. Parts of this section have been published in [85].

9.2.1 Reverse Fuzzy Extractors

Fuzzy extractors and secure sketches (cf. Section 6.3) are commonly used to correct noisy PUF responses on the PUF-enabled device, which is required when the PUF response is used in a cryptographic algorithm or protocol [68, 56, 184, 4]. However, the underlying error decoding algorithms are typically complex and require a large number of gates and/or long execution times when multiple bit errors must to be corrected [57, 26]. Hence, implementing the decoding algorithm on the PUF-enabled device is a huge disadvantage in many applications.

To overcome this problem, we use *reverse fuzzy extractors* [85] that allow for very compact and fast implementations of secure sketches and fuzzy extractors. Reverse fuzzy extractors use the much more efficient helper data generation phase **FEGen** on the PUF-enabled device instead of the computationally intensive reproduction phase **FERep** and **FERep** is moved to the typically more powerful reader (cf. Figure 9.1). As a consequence, a new helper data h is generated each time the PUF is queried and the reader corrects the reference value y in its database to the actual PUF response y' , which is different each time the PUF is evaluated due to environmental variations.

There is one major pitfall that must be considered: Each execution of the helper data generator **FEGen** on a different noisy version of the same PUF response reveals new helper data. However, secure sketches give no guarantee about the min-entropy of

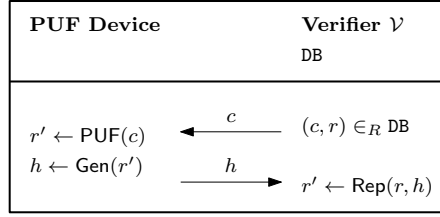


Figure 9.1: Concept of Reverse Fuzzy Extractors

the PUF response in case *multiple* helper data for different noisy variants of the same response are known [27]. Hence, reverse fuzzy extractors may leak the full PUF response, when FEGen and FERep are based on a conventional fuzzy extractor. This is problematic in most PUF-based applications, such as controlled PUFs and PUF-based key storage (cf. Sections 6.6.1 and 6.4.2) that require at least some bits of the PUF response to be secret and must be carefully considered when designing reverse fuzzy extractors.

Reverse fuzzy extractors have been implemented [85] based on the syndrome construction [57], which is a secure sketch with a highly efficient helper data generation phase and that has been shown to ensure a certain amount of min-entropy in the PUF response, even if multiple helper data for noisy variants of the same PUF response are known [27]. The syndrome construction implements the helper data generator FEGen(y) as $h \leftarrow y \cdot H^T$, where H is the parity check matrix of a binary linear block code and h corresponds to the syndrome of y . The reproduction algorithm FERep(y', h) of the syndrome construction computes $y \leftarrow y' - e$, where e is determined by decoding the syndrome $s = h - y' \cdot H^T$ using the decoding algorithm of the underlying error correcting code. Note that FEGen corresponds to computing a matrix product of the PUF response with the parity-check matrix H of the underlying cyclic linear block code. Due to the special form of parity check matrices of these codes, this product can be computed very efficiently [85].

9.2.2 Protocol Description and Specification

A naive approach to authenticate a PUF-enabled RFID tag is the following: The reader sends a random PUF challenge from a reference database to the tag and accepts the tag only when its response is similar to the one in the database. However, since the tag always responds to the same PUF challenge with a similar PUF response, replay attacks are possible. Moreover, for most PUF implementations, sending the PUF response in clear allows cloning the tag by model building attacks [160]. Further, it is not trivial to

authenticate the reader to the tag following this approach.

Our scheme solves these problems by merging the idea of controlled PUFs [68] and logically reconfigurable PUFs [108]: We amend a PUF with a control logic that (1) hides the plain PUF response from the adversary and (2) allows dynamically changing the challenge/response behavior of the PUF in a random manner. Using reverse fuzzy extractors allows for a very compact implementation of our scheme that requires only minimal resources on the tag.

System Model

The players in our scheme are (at least) a tag issuer \mathcal{I} , a reader \mathcal{R} and a tag \mathcal{T} . We denote the adversary with \mathcal{A} . Our scheme enables *mutual authentication* between \mathcal{R} and \mathcal{T} . \mathcal{R} has access to a database DB containing detailed information on all tags \mathcal{T} in the system. DB is initialized and maintained by \mathcal{I} .

Trust Model and Assumptions

Issuer \mathcal{I} and reader \mathcal{R} . We assume \mathcal{I} and \mathcal{R} to be trusted, which is a typical assumption in most RFID systems (cf. Section 3.2).¹ Further, \mathcal{I} initializes \mathcal{T} and \mathcal{R} in a secure environment.

Tag \mathcal{T} . We consider \mathcal{T} to be a passive device that cannot initiate communication, has a narrow communication range (a few centimeters to meters) and erases its temporary state (all session-specific information and randomness) after it gets out of the electromagnetic field of \mathcal{R} . Further, we assume \mathcal{T} to be equipped with a robust and unpredictable PUF (cf. Section 6.1), a reverse fuzzy extractor (cf. Section 9.2.1) and a lightweight hash function.

Adversary \mathcal{A} . As in most RFID security models, we assume \mathcal{A} to control the wireless communication channel between \mathcal{R} and \mathcal{T} (cf. Section 3.2). This means that \mathcal{A} can eavesdrop, manipulate, delete and reroute all protocol messages sent by \mathcal{R} and \mathcal{T} . Moreover, \mathcal{A} can obtain useful information (e.g., by visual observation) on whether \mathcal{R}

¹Note that there are papers considering revocation of malicious readers [12, 139]. A simple approach to enable reader revocation in our scheme is moving all computations of \mathcal{R} to DB such that \mathcal{R} has no access to the PUF challenge/response pairs.

accepted \mathcal{T} as a legitimate tag. Following the typical assumptions on PUF-based key storage [195, 120, 184], we assume that \mathcal{A} can read any information that is stored in the non-volatile memory of \mathcal{T} . However, \mathcal{A} cannot access the responses of the PUF of \mathcal{T} and cannot obtain temporary data stored in the volatile memory (such as intermediate results of the computations) of \mathcal{T} while it is participating in an authentication protocol. This can be achieved by using side-channel aware designs for the implementation of the underlying algorithms.

Protocol Specification

System initialization. The tag issuer \mathcal{I} stores a random tag identifier ID in the non-volatile memory of the tag \mathcal{T} . Moreover, \mathcal{I} extracts $q > 0$ challenge/response pairs $(x_1, y'_1), \dots, (x_q, y'_q)$ from the PUF of \mathcal{T} and stores them together with ID in the database DB, which is later used by the reader \mathcal{R} in the authentication protocol.

Authentication protocol. The authentication protocol is depicted in Figure 9.2 and works as follows: Verifier \mathcal{R} starts by sending an authentication request **auth** to the tag \mathcal{T} , which responds with its identifier ID . \mathcal{R} chooses a random nonce N and a random challenge/response pair (x_i, y'_i) from the database DB and sends (x_i, N) to \mathcal{T} . Then, \mathcal{T} evaluates $y_i \leftarrow \text{PUF}(x_i)$, generates $h_i \leftarrow \text{FEGen}(y_i)$ using the reverse fuzzy extractor, computes $a \leftarrow \text{Hash}(ID, N, y_i, h_i)$ and sends (h_i, a) to \mathcal{R} . Next, \mathcal{R} reproduces $y_i \leftarrow \text{FERep}(y'_i, h_i)$ using y'_i from DB and checks whether $\text{Hash}(ID, N, y_i, h_i) = a$. If this is not the case, \mathcal{R} aborts and rejects. Otherwise, \mathcal{R} sends $b \leftarrow \text{Hash}(a, y_i)$ to \mathcal{T} and accepts. Eventually, \mathcal{T} accepts if $\text{Hash}(a, y_i) = b$ and rejects otherwise.

Discussion. Note that the case $q = 1$ is equivalent to PUF-based key storage, where y_1 represents the authentication secret of \mathcal{T} . In this case, x_1 can be stored in the non-volatile memory of \mathcal{T} and needs not to be sent from \mathcal{R} to \mathcal{T} . Hence, two protocol messages can be saved: N can be sent with **auth** and ID can be sent with (h_i, a) . Using multiple challenge/response pairs corresponds to storing multiple (session) keys in the PUF, which limits the impact of side channel attacks that may recover only a subset of these keys.

Further, note that although the helper data h_i does not leak enough information to recover the complete PUF-response y_i (cf. Section 9.2.4), it may still leak enough infor-

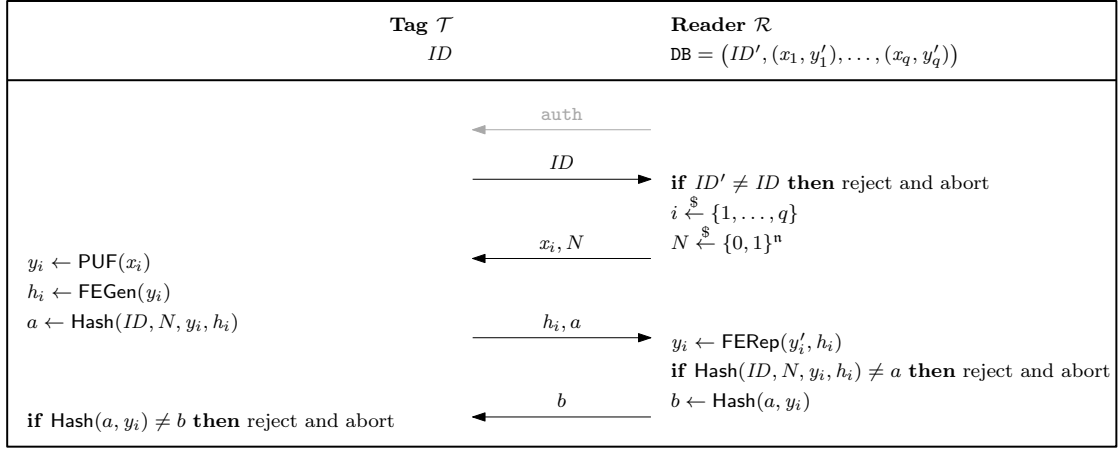


Figure 9.2: Lightweight PUF-based Mutual Authentication Protocol

mation on y_i to trace the tag.

9.2.3 Performance Evaluation

The feasibility of the reverse fuzzy extractor and the efficiency of the protocol design has been shown by a prototype implementation of the protocol [85]. The prototype is designed to be used with an Arbiter PUF but can be easily modified to work with most existing intrinsic PUFs. The Arbiter PUF implementation used accepts 64-bit challenges and generates 1-bit responses. Since our protocol requires multiple response bits, a linear feedback shift register (LFSR) is used to expand a single challenge x into many consecutive 64-bit challenges x' , which are fed one after the other into the PUF. This allows generating PUF responses of arbitrary length for a single challenge. The expansion can be omitted for other PUF types that generate responses that have a sufficient length.

As described in Section 9.2.1, the syndrome generation consists of the matrix multiplication of an n -bit PUF response y with the $n \times (n - k)$ parity check matrix H^T of an error correcting linear block code. This $(n - k)$ -bit result is called helper data h for y and can be used to correct a noisy version of y if the number of bit errors is small enough. The implementation uses the parity-check matrix of a $[n = 255, k = 21, t = 55]$ BCH block code that can correct up to $t = 55$ erroneous bits in a $n = 255$ bit PUF response and uses a $n - k = 234$ -bit helper data vector. In case the probability of a single bit error is 10% (cf. Section 7.4.1), then the probability of observing more than 55 errors in 255 response

bits (resulting in a decoding failure) will only happen with probability $10^{-7.82}$. Due to the special structure of parity-check matrices of BCH codes, the matrix multiplication can be efficiently implemented as an linear feedback shift register (LFSR) [85].

The helper data must be sent from the tag to the reader, which causes an entropy loss of the actual PUF response of up to $n - k$ bits. Assuming the PUF response has full entropy, there will be only $k = 21$ bits of uncertainty left after observing the helper data. In order to obtain a security level equivalent to a 128-bit key, we need at least $\lceil 128/21 \rceil = 7$ responses, each 255 bits in length and 7 corresponding helper data vectors. This leads to an overall PUF response length of $7 \cdot 255 = 1,785$ bits and an overall helper data length of $7 \cdot 234 = 1,638$ bits. The probability of an authentication failure due to a decoding failure in one of the 7 blocks is $1 - (1 - 10^{-7.82})^7 = 10^{-6.97}$. This means that the implementation of the protocol achieves a false rejection rate of only one in approximately 10 million authentications. The prototype uses the lightweight hash function SPONGENT [22] which seems to be perfectly suited for resource-constrained tags.

When synthesized for a Xilinx Virtex-5 FPGA, the complete tag-side implementation of the protocol (except the PUF) uses only 496 one-bit flip-flops and 658 6-input lookup-tables (LUTs) [85].

9.2.4 Security Analysis

We now prove the security properties of the reverse fuzzy extractor construction and the mutual authentication scheme. In this context, we formalize all necessary aspects and set up formal security definitions.

Security of the Reverse Fuzzy Extractor

Secure sketch. Let \mathbf{M} be a metric space with n elements and distance metric dist . Moreover, let $\mathbf{C} = \{w_1, \dots, w_k\} \subseteq \mathbf{M}$ be an error correcting code with codewords w_i for $1 \leq i \leq k$. Let d be the minimum distance and t be the error correcting distance of \mathbf{C} , which means that \mathbf{C} can detect up to d and correct up to t errors. We only consider linear binary block codes, where $\mathbf{M} = \mathbb{F}_2^n$ and dist corresponds to the Hamming distance. These codes are commonly denoted as $[n, k, d]$ codes and it holds that $t = \lfloor (d - 1)/2 \rfloor$. Following [57], we formally define a secure sketch as follows:

Definition 9.1 (Secure Sketch). A (\mathbb{M}, m, m', t) -secure sketch is a pair of probabilistic polynomial time algorithms **FEGen** and **FERep** with the following properties: **FEGen** takes input $w \in \mathbb{M}$, which is chosen according to a distribution W on \mathbb{M} and returns a bit-string $h \in \{0, 1\}^*$. **FERep** takes inputs $w' \in \mathbb{M}$ and h and returns $w'' \in \mathbb{M}$. Correctness ensures that $w'' = w$ if $h = \text{FEGen}(w)$ and $\text{dist}(w, w') \leq t$. The security property guarantees that for any distribution W over \mathbb{M} with min-entropy m , w can be recovered from (a single) $h = \text{FEGen}(w)$ with at most probability $2^{-m'}$.

Next, we specify the syndrome construction that has been informally discussed in Section 9.2.1 and that has been shown to implement a secure sketch [57]:

Definition 9.2 (Syndrome Construction). The syndrome construction is a $(\mathbb{F}_2^n, n, k, t)$ -secure sketch (Definition 9.1) that is based on a linear binary $[n, k, d]$ error correcting block code. **FEGen**(w) computes $h \leftarrow w \cdot H^T$, where H is the parity check matrix of the underlying code. **FERep**(w', h) computes $w \leftarrow w' - e$, where e is determined by decoding the syndrome $s = h - w' \cdot H^T$ using the decoding algorithm of the underlying code.

Note that the helper data $h = w \cdot H^T$ corresponds to the syndrome of w . However, since the syndrome construction does not require w to be a codeword, decoding h may most likely fail. To overcome this problem, the reproduction algorithm **FERep** of the syndrome construction decodes the syndrome $s = h - w' \cdot H^T = e \cdot H^T$.

Security definition of reverse fuzzy extractors. Similar to conventional fuzzy extractors, reverse fuzzy extractors should ensure that the helper data does not leak the full PUF response. However, for reverse fuzzy extractors this must hold even when *multiple* different helper data for noisy variants of the *same* PUF response are known. This has been formalized by Boyen [27] as *outsider chosen perturbation security*, which is defined based on a security experiment between an unbounded adversary \mathcal{A} and a challenger \mathcal{C}_{PS} . In this experiment, \mathcal{A} interacts with the helper data generator **FEGen** of a secure sketch (Definition 9.1) and obtains the helper data for different $w_i = w + e_i$ for a fixed but secret w and different noise vectors (perturbations) e_i that can be adaptively chosen by \mathcal{A} . This allows \mathcal{A} to influence the noise, which in the case of PUFs can be done by changing the operating conditions such as the ambient temperature or the supply voltage. The outsider chosen perturbation security experiment is defined as follows: \mathcal{A} sends a description of distribution W over \mathbb{M} to \mathcal{C}_{PS} , which then samples $w \in \mathbb{M}$ according

to W . Next, \mathcal{A} interacts with FEGen and obtains an arbitrary number of helper data $h_i = \text{FEGen}(w_i)$ for different $w_i = w + e_i$, where $e_i \in \mathbb{M}$ can be adaptively chosen by \mathcal{A} with the only restriction that the Hamming weight of e_i is less or equal to t . Eventually, \mathcal{A} returns a guess w^* for w . \mathcal{A} wins if $w^* = w$. Based on this security experiment, Boyen [27] sets up the following security definition:

Definition 9.3 (Chosen Perturbation Security). *A (\mathbb{M}, m, m', t) -secure sketch (Definition 9.1) is unconditionally secure against adaptive outsider chosen perturbation attacks if no unbounded adversary \mathcal{A} can win the outsider chosen perturbation security experiment described above with probability greater than $2^{-m'}$ for any distribution W over \mathbb{M} with min-entropy m .*

Moreover, Boyen [27] shows that the syndrome construction achieves outsider chosen perturbation security:

Theorem 9.1 (Chosen Perturbation Security of the Syndrome Construction). *The syndrome construction (Definition 9.2) is unconditionally secure against adaptive outsider chosen perturbation attacks (Definition 9.3).*

We now state the security of the reverse fuzzy extractor construction:

Theorem 9.2 (Chosen Perturbation Security of the Reverse Fuzzy Extractor). *The reverse fuzzy extractor (Section 9.2.1) based on the syndrome construction (Definition 9.2) is a $(\mathbb{F}_2^n, n, k, t)$ -secure sketch (Definition 9.1) that achieves outsider perturbation security (Definition 9.3).*

Proof of Theorem 9.2. Note that FEGen and FERep of the syndrome construction and the the reverse fuzzy extractor based on the syndrome construction are identical. In fact, only the entities that execute FEGen and FERep have been switched. Hence, the syndrome construction and the reverse fuzzy extractor based on the syndrome construction are equivalent. Thus, since the syndrome construction is a $(\mathbb{F}_2^n, n, k, t)$ -secure sketch, the reverse fuzzy extractor based on the syndrome construction is also a $(\mathbb{F}_2^n, n, k, t)$ -secure sketch. Consequently, it follows from Theorem 9.1 that the reverse fuzzy extractor based on the syndrome construction achieves outsider perturbation security. \square

Security of the Authentication Protocol

Correctness. Correctness means that, in case tag \mathcal{T} and reader \mathcal{R} are honest, mutual authentication should be successful.

Definition 9.4 (Correctness). *A mutual authentication scheme is correct, if an honest \mathcal{T} always makes an honest \mathcal{R} accept and an honest \mathcal{R} always makes an honest \mathcal{T} accept.*

Theorem 9.3 (Correctness). *The authentication scheme in Section 9.2.2 is correct, when based on a $(\mathbb{F}_2^n, n, k, t)$ -secure sketch (Definition 9.1) and a PUF that generates responses of length n bits with at most t bit errors.*

Proof of Theorem 9.3. It is easy to see that the protocol in Section 9.2.2 is correct if $\text{FERep}(y'_i, \text{FEGen}(y_i)) = y_i$ for all (y_i, y'_i) . The correctness property of the $(\mathbb{F}_2^n, n, k, t)$ -secure sketch (Definition 9.1) ensures that $\text{FERep}(y'_i, \text{FEGen}(y_i)) = y_i$ if $\text{dist}(y_i, y'_i) \leq t$. If the PUF generates responses of length n bits with a bit error rate of at most ρ , then the probability of $\text{dist}(y, y') \leq t$ can be expressed as the cumulative binomial distribution in t with parameters ρ and n . Note that t is chosen such that this probability, which is an upper bound for the false rejection rate of the authentication, becomes very small. Hence, a $(\mathbb{F}_2^n, n, k, t)$ -secure sketch can then recover y from y' with overwhelming probability. \square

Note that the implementation described in Section 9.2.3 can handle PUFs with $\rho \leq 10\%$. When both reader and tag are trusted, it achieves an authentication failure rate of less than $10^{-6.97}$, which is acceptable for most commercial applications.

Tag authentication. Tag authentication means that the adversary \mathcal{A} should not be able to make a legitimate reader \mathcal{R} accept. Similar as in Sections 5.4.4 and 5.3.4, we formalize tag authentication based on a security experiment where \mathcal{A} must make an honest \mathcal{R} to authenticate \mathcal{A} as \mathcal{T} . Hereby, \mathcal{A} can arbitrarily interact with \mathcal{T} and \mathcal{R} , which both are simulated by a challenger \mathcal{C}_{TA} . However, since in general it is not possible to prevent simple relay attacks, \mathcal{A} is not allowed to just forward all messages from \mathcal{T} to \mathcal{R} .² This means that at least some of the protocol messages that made \mathcal{R} accept must have been computed by \mathcal{A} . More specifically, the tag authentication experiment is as follows: \mathcal{C}_{TA} initializes \mathcal{T} and \mathcal{R} . Then, \mathcal{C}_{TA} initializes \mathcal{A} with the public system parameters. Next, \mathcal{A} can arbitrarily interact with \mathcal{T} and \mathcal{R} that are simulated by \mathcal{C}_{TA} . Hereby, \mathcal{A} can eavesdrop on authentication protocol-runs between an honest \mathcal{T} and an honest \mathcal{R} and manipulate protocol messages exchanged between \mathcal{R} and \mathcal{T} . Further, \mathcal{A}

²Note that simple relay attacks can be mitigated by distance bounding techniques. However, for simplicity we excluded relay attacks because the main focus of the protocol is demonstrating the use of reverse fuzzy extractors.

can start authentication protocol-runs as \mathcal{R} or \mathcal{T} with \mathcal{C}_{TA} . \mathcal{A} wins, if he makes \mathcal{R} accept after a polynomial (in the bit length l of the nonces) number of queries to \mathcal{C}_{TA} .

Definition 9.5 (Tag Authentication). *An authentication scheme achieves μ -tag authentication, if no probabilistic polynomial time adversary \mathcal{A} wins the tag authentication experiment with probability greater than $2^{-\mu}$.*

Theorem 9.4 (Tag Authentication). *The authentication scheme in Section 9.2.2 achieves k -tag authentication (Definition 9.5) in the random oracle model, when using the reverse fuzzy extractor (Section 9.2.1) based on the syndrome construction (Definition 9.2).*

In the following, we focus on the variant of our authentication scheme that uses only one single challenge/response pair, i.e., where $q = 1$. The proof can be easily extended for $q > 1$.

Proof of Theorem 9.4. We show that, if there is an adversary \mathcal{A} who violates tag authentication (Definition 9.5) with probability greater than 2^{-k} , then \mathcal{A} can be transformed into an adversary \mathcal{A}' who violates the outsider chosen perturbation security of the reverse fuzzy extractor (Theorem 9.2). Note that, in the chosen perturbation security experiment (Definition 9.3), \mathcal{A}' interacts with a helper data generator oracle FEGen that, when queried with e_j , returns $h_j = \text{FEGen}(y + e_j)$ for a fixed but unknown $y \in \mathbb{F}_2^n$. Based on this FEGen -oracle, \mathcal{A}' simulates the challenger \mathcal{C}_{TA} of the tag authentication security experiment (Definition 9.5) such that \mathcal{A} cannot distinguish between \mathcal{A}' and \mathcal{C}_{TA} . Hereby, \mathcal{A} and \mathcal{A}' have access to the same random oracle Hash and \mathcal{A}' records all queries x made by \mathcal{A} to Hash and the corresponding responses $\text{Hash}(x)$ in a list L . Since \mathcal{A} cannot distinguish \mathcal{A}' from \mathcal{C}_{TA} , by assumption \mathcal{A} violates tag authentication (Definition 9.5) with probability greater than 2^{-k} . \mathcal{A}' uses L to extract y from the protocol message (h, a) generated by \mathcal{A} that finally makes \mathcal{R} accept. Note that the random oracle ensures that $(x, a) \in L$. Hence, \mathcal{A}' can extract y with probability greater than 2^{-k} , which contradicts the outsider chosen perturbation security of the reverse fuzzy extractor (Theorem 9.2). \square

Note that in practice, the success probability $2^{-\mu}$ (Definition 9.5) of \mathcal{A} may depend on the output length t of the hash function implementing the random oracle. In case $t < k$, \mathcal{A} could simply guess the correct hash digest a with probability 2^{-t} . For the implementation of the reverse fuzzy extractor based on the syndrome construction discussed in Section 9.2.3, we have $t = 128 < k = 147$ and thus $\mu = 128$.

Verifier authentication. Verifier authentication means that the adversary \mathcal{A} should not be able to make an honest tag \mathcal{T} accept. This is formalized by a reader authentication security experiment between \mathcal{A} and a challenger \mathcal{C}_{RA} that is identical to the tag authentication experiment with the only difference that \mathcal{A} wins, if \mathcal{A} makes \mathcal{T} accept after a polynomial (in the output bit length t of Hash and the bit length of the PUF responses) number of queries.

Definition 9.6 (Reader Authentication). *An authentication scheme achieves μ -reader authentication, if no probabilistic polynomial time adversary \mathcal{A} wins the reader authentication experiment with probability greater than $2^{-\mu}$.*

Theorem 9.5 (Reader Authentication). *The authentication scheme in Section 9.2.2 achieves k -reader authentication (Definition 9.6) in the random oracle model, when using the reverse fuzzy extractor (Section 9.2.1) based on the syndrome construction (Definition 9.2) when the underlying PUF generates at least ρ bit errors each time it is evaluated.*

Proof of Theorem 9.5. We show that, if there is an adversary \mathcal{A} who violates reader authentication (Definition 9.6) with probability greater than 2^{-k} , then \mathcal{A} can be transformed into an adversary \mathcal{A}' who violates the outsider chosen perturbation security of the reverse fuzzy extractor (Theorem 9.2). \mathcal{A}' simulates challenger \mathcal{C}_{RA} of the reader authentication security experiment (Definition 9.5) based on the FEGen-oracle such that \mathcal{A} cannot distinguish between \mathcal{A}' and \mathcal{C}_{RA} in a similar way as in the proof of Theorem 9.4. Hereby, \mathcal{A} and \mathcal{A}' have access to the same random oracle and \mathcal{A}' records all queries x made by \mathcal{A} to Hash and the corresponding responses $\text{Hash}(x)$ in a list L . Since \mathcal{A} cannot distinguish between \mathcal{A}' and \mathcal{C}_{RA} , by assumption \mathcal{A} violates reader authentication (Definition 9.6) with probability greater than 2^{-k} . \mathcal{A}' uses L to extract y from the protocol message b generated by \mathcal{A} that finally makes \mathcal{T} accept. Note that the random oracle assumption ensures that $(x, b) \in L$, while the bit errors in the PUF responses ensure that \mathcal{A} cannot just replay an old hash digest b . Hence, \mathcal{A}' can extract y with probability greater than 2^{-k} , which contradicts the outsider chosen perturbation security of the reverse fuzzy extractor (Theorem 9.2). \square

9.2.5 Conclusion

We presented a new lightweight PUF-based authentication scheme providing mutual authentication of RFID tags and readers. Our scheme is resistant to emulation attacks, sup-

ports an unlimited number of tag authentications and does not require the reader to store a large number of PUF challenge/response pairs. Our protocol is based on the concept of *reverse fuzzy extractors* [85], a novel approach to correct noise in PUF responses moving the computationally expensive error correction process from the resource-constrained PUF-enabled tag to the more powerful RFID reader. Reverse fuzzy extractors are applicable to device authentication and PUF-based key storage (where the key is used to communicate with an external entity) and can significantly reduce the area costs of secure PUF implementations.

9.3 Privacy-preserving PUF-based RFID Authentication

While PUFs allow to uniquely identify devices, we show that they can also be used to enhance the privacy in RFID systems. In this section, we present a privacy-preserving tag authentication protocol for RFID that can be proven to be destructive private in a variant of the RFID security and privacy model by Vaudenay [190] (cf. Section 5.3.4), which we denote as V-Model in the following. This means that our protocol provides untraceability of tags against adversaries that permanently destroy a tag by physically attacking (i.e., corrupting) it. Our protocol is based on the weak private protocol proposed by Vaudenay [190] and uses Physically Unclonable Functions (PUFs) as tamper-evident key storage in a similar way as described by Tuyls et al. [184]. This means that the tag authentication key is not stored on the tag but reconstructed from the physical characteristics of the RFID chip each time it is needed. The properties of the PUF ensure that any attempt to physically tamper with the PUF to obtain the authentication secret of the tag results in destruction of the PUF and the tag secret, which corresponds to the definition of a destructive adversary in the V-Model. According to Tuyls et al. [184], such a PUF-based key storage (including the PUF and the required error correction mechanism) can be implemented with less than 1,000 gates, which is well within the capabilities of common RFID tags.

Note that in the V-Model, the only information that differentiates a tag from another tag is a binary state S that is stored on each tag during its creation. However, the use of PUFs implies placing on the tag a physical (non-digital) object, thus differentiating tags also from a physical point of view. Hence, we solve the problem of achieving destructive privacy in a very mild variant of the V-Model that includes the possibility of physically differentiating tags during their creation, in our case through the use of PUFs. For the

sake of simplifying the exposition in the remaining part of the paper we will not insist in claiming the *revisited* V-Model, but we will stick with V-Model.

Remark. The PUF-based authentication protocol and its security analysis are due to the author of this work and the result of many intensive discussions with Ahmad-Reza Sadeghi (TU Darmstadt, Germany) and Ivan Visconti (University of Salerno, Italy). Parts of this section have been published in [168, 167].

9.3.1 Protocol Description and Specification

Our destructive-private RFID protocol is based on the weak-private protocol by Vaude- nay [190], which is a simple challenge-response protocol. To achieve destructive-privacy, in our protocol, the tag \mathcal{T} does not directly use its state S as authentication key K . Instead, K is derived by evaluating a Physically Unclonable Function PUF on input S each time K is needed. Hence, the properties of the PUF ensure that the adversary cannot access the tag secret K but destroys the tag \mathcal{T} by any attempt to corrupt it. For simplicity, we consider an ideal PUF since the focus of this section is showing the feasibility of destructive-private authentication.

Definition 9.7 (Ideal PUF). *Let $l \in \mathbb{N}$ be a security parameter, $\gamma, \kappa \in \mathbb{N}$ be polynomially bounded in l and $\text{PUF} : \{0, 1\}^\gamma \rightarrow \{0, 1\}^\kappa$ be an ideal Physically Unclonable Function (PUF). Consider the following security experiment $\text{Exp}_{\mathcal{A}_{\text{puf}}}^{\text{puf-b}}$, where an adversary \mathcal{A}_{puf} interacts with a PUF-challenger \mathcal{C}^{puf} : When initialized with l, γ, κ and $b \xleftarrow{\$} \{0, 1\}$, the PUF-challenger \mathcal{C}^{puf} initializes an oracle \mathcal{O}^{PUF} that on input $x \in \{0, 1\}^\gamma$ returns $y \leftarrow \text{PUF}(x)$ if $b = 1$ and $y \xleftarrow{\$} \{0, 1\}^\kappa$ otherwise. After a polynomial number of queries to \mathcal{O}^{PUF} , \mathcal{A}_{puf} must return a bit b' . \mathcal{A}_{puf} wins the security experiment if $b = b'$. An ideal PUF is a function PUF with the following properties:*

1. *For all $x \in \{0, 1\}^\gamma$ and all $(y_i, y_j) \in [\text{PUF}(x)]^2$ it holds that $\Pr[y_i = y_j] = 1$.*
2. *Each p.p.t. adversary \mathcal{A}_{puf} has at most negligible advantage*

$$\text{Adv}_{\mathcal{A}_{\text{puf}}}^{\text{puf}} = |\Pr[\text{Exp}_{\mathcal{A}_{\text{puf}}}^{\text{puf-1}} = 1] - \Pr[\text{Exp}_{\mathcal{A}_{\text{puf}}}^{\text{puf-0}} = 1]|.$$

3. *Any attempt to physically tamper with the object implementing PUF results in destruction of PUF, i.e., PUF cannot be evaluated any more.*

Note that the second property in Definition 9.7 is similar to the pseudo-randomness property of a PRF (cf. Definition 2.4). Hence, the output of an ideal PUF is pseudo-random, which can be achieved by using fuzzy extractors (cf. Section 6.3). In addition, the second property in Definition 9.7 implies that the adversary cannot compute the output of the PUF for an adaptively chosen challenge even after adaptively querying the PUF for a polynomial number of times. This implies that the adversary cannot emulate (i.e., impersonate or clone) the PUF based on its input/output behaviour, which seems to be achievable by using, e.g., Controlled PUFs (cf. Section 6.6.1). The third property in Definition 9.7 ensures that the adversary cannot obtain any information on the PUF by physical means, which prevents cloning the PUF. Moreover, the capabilities of the adversary are not limited concerning the creation and querying of other PUFs, which means that different ideal PUFs are independent pseudo-random functions.

Assumptions. Following existing work on PUF-based authentication of RFID tags [184, 31, 103] (cf. Section 6.4.2), we assume that the adversary cannot access the PUF-responses. This can be achieved, e.g., by implementing the algorithm processing the PUF-response in hardware and in a side-channel resilient way.

Protocol specification. Let $l \in \mathbb{N}$ be a given security parameter, $\alpha, \beta, \gamma, \kappa \in \mathbb{N}$ be polynomial in l and let $F : \{0, 1\}^\kappa \times \{0, 1\}^{2\alpha} \rightarrow \{0, 1\}^\beta$ be a family of pseudo-random functions.³ Each tag \mathcal{T} is equipped with a (unique) ideal Physically Unclonable Function $\text{PUF} : \{0, 1\}^\gamma \rightarrow \{0, 1\}^\kappa$ (Definition 9.7) and is initialized by a random state $S \xleftarrow{\$} \{0, 1\}^\gamma$. The credentials database DB of the reader \mathcal{R} contains a tuple (ID, K) for each legitimate tag \mathcal{T} where $K \leftarrow \text{PUF}(S)$.

The destructive-private tag authentication protocol is shown in Figure 9.3 and works as follows: \mathcal{R} starts by sending a random challenge a to \mathcal{T} , which first chooses a random b and then queries PUF with S to reconstruct K . Next, \mathcal{T} evaluates $F_K(a, b)$, sends the result c and b to \mathcal{R} and immediately erases K , a , b and c from its temporary memory. On receipt of c , \mathcal{R} recomputes $F_K(a, b)$ for each tuple (ID, K) in DB until it finds a match. If \mathcal{R} finds a matching (ID, K) , it accepts \mathcal{T} by returning ID . Otherwise, \mathcal{R} rejects \mathcal{T} and returns \perp .

³Note that in practice F could be instantiated as a Message Authentication Code (MAC).

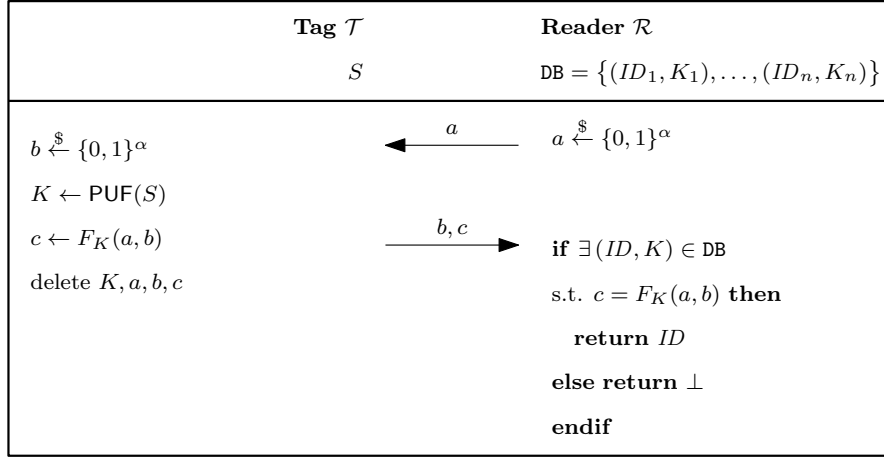


Figure 9.3: Destructive-Private PUF-based Tag Authentication Protocol

Correctness. Clearly, if both \mathcal{T} and \mathcal{R} are legitimate, then the correctness (Definition 4.2) of the tag authentication protocol in Figure 9.3 follows directly from the properties of F (Definition 2.4) and PUF (Definition 9.7).

9.3.2 Security Analysis

We analyse the security and privacy properties of our destructive-private authentication scheme in the V-Model. The V-Model is very similar to the PV-Model (cf. Section 4.2) with the only difference that the V-Model does not include reader authentication. Specifically the differences of the V-Model to the PV-Model are that (1) in the **AuthTag** protocol (cf. Definition 4.1), the tag \mathcal{T} does not produce any output and (2) there is no definition of reader authentication, i.e., Definition 4.5 does not exist in the V-Model.

Theorem 9.6 (Tag Authentication). *The RFID protocol illustrated in Figure 9.3 achieves tag authentication (Definition 4.4) if F is a PRF (Definition 2.4).*

Proof of Theorem 9.6. Assume by contradiction that there is an adversary \mathcal{A}_{sec} against the protocol shown in Figure 9.3 who violates tag authentication (Definition 4.4). We show that \mathcal{A}_{sec} can be transformed into an algorithm \mathcal{A}_{prf} that contradicts the security property of F (Definition 2.4). The main idea of the proof is as follows:

\mathcal{A}_{prf} uses \mathcal{O}^{F_K} to simulate the oracles defined in Section 4.2.2 to \mathcal{A}_{sec} . After a polynomial number of interactions with the oracles, \mathcal{A}_{sec} returns a *new* protocol message

(\tilde{b}, \tilde{c}) for a given value \tilde{a} . Note that \mathcal{A}_{sec} is not allowed to make a $\text{SendTag}(\tilde{a}, \cdot)$ query to the tag \mathcal{T} , which ensures that $\mathcal{O}^{F_{\tilde{K}}}$ has not been queried with (\tilde{a}, \tilde{b}) before. Now, \mathcal{A}_{prf} sends $x \leftarrow (\tilde{a}, \tilde{b})$ to \mathcal{C}^{prf} who responds with a challenge y . Note that in case $\mathcal{O}^{F_{\tilde{K}}}$ simulates $F_{\tilde{K}}$, the simulation of the oracles to \mathcal{A}_{sec} is perfect. Hence, in this case, by assumption with non-negligible probability it holds that $\tilde{c} = F_{\tilde{K}}(\tilde{a}, \tilde{b})$. This means that if $\mathcal{O}^{F_{\tilde{K}}}$ simulates $F_{\tilde{K}}$, then $\tilde{c} = y$ must hold with non-negligible probability. Clearly, this allows \mathcal{A}_{prf} to distinguish between $F_{\tilde{K}}$ and a randomly chosen value, which contradicts the pseudo-randomness of F (Definition 2.4). The detailed proof is as follows:

Assume by contradiction that the protocol shown in Figure 9.3 does not achieve tag authentication. This means that there is an adversary \mathcal{A}_{sec} who can generate, with non-negligible probability p , a protocol message (\tilde{b}, \tilde{c}) for a given \tilde{a} such that $\tilde{c} = F_{\tilde{K}}(\tilde{a}, \tilde{b})$ where $(\tilde{ID}, \tilde{K}) \in \text{DB}$ without having made a CorruptTag or $\text{SendTag}(\tilde{a}, \cdot)$ query to the tag \mathcal{T} . In the following, we show that \mathcal{A}_{sec} can be transformed into a probabilistic polynomial time algorithm \mathcal{A}_{prf} that contradicts the security property of the underlying PRF F (Definition 2.4). Hence, the pseudo-randomness of F ensures that there is no such adversary \mathcal{A}_{sec} .

The construction of \mathcal{A}_{prf} is as follows: Given the security parameters l, κ, α, β and a description of the PRF F from the PRF-challenger \mathcal{C}^{prf} , \mathcal{A}_{prf} initializes the RFID system by first choosing γ polynomial in l and then setting $sk_{\mathcal{R}} \leftarrow \emptyset, pk_{\mathcal{R}} \leftarrow (l, \gamma, \kappa, \alpha, \beta, F)$ and $\text{DB} \leftarrow \{\}$. Then \mathcal{A}_{prf} guesses the identifier \tilde{ID} of the tag \mathcal{T} that will be impersonated by \mathcal{A}_{sec} . Note that the probability of correctly guessing \tilde{ID} is polynomial since \mathcal{A}_{sec} can create at most a polynomial number of tags. Next, \mathcal{A}_{prf} initializes \mathcal{A}_{sec} with $(l, \gamma, \kappa, \alpha, \beta, F)$ and simulates all the oracles defined in Section 4.2.2 to \mathcal{A}_{sec} :

CreateTag(ID) If there already is a tuple $(ID, \cdot, \cdot) \in \text{DB}$ or if $ID = \tilde{ID}$, then \mathcal{A}_{prf} aborts.

Otherwise, \mathcal{A}_{prf} chooses $S \xleftarrow{\$} \{0, 1\}^\gamma$ and $K \xleftarrow{\$} \{0, 1\}^\kappa$ and updates $\text{DB} \leftarrow \text{DB} \cup \{(ID, K, S)\}$.

DrawTag, FreeTag, LaunchIdent The simulation of the **DrawTag**, **FreeTag** and **LaunchIdent** oracle is straightforward. Note that \mathcal{A}_{prf} knows the secret look-up table Γ of the **DrawTag** oracle.

SendTag($a, vtag$) If $\Gamma[vtag] = \tilde{ID}$, \mathcal{A}_{prf} responds with $b \xleftarrow{\$} \{0, 1\}^\alpha$ and $c \leftarrow \mathcal{O}^{F_{\tilde{K}}}(a, b)$.

Else, \mathcal{A}_{prf} gets $(\Gamma[vtag], K, S)$ from DB and responds with $b \xleftarrow{\$} \{0, 1\}^\alpha$ and $c \leftarrow F_K(a, b)$.

- SendReader**(\emptyset, π) If π has been previously generated by a **LaunchIdent** oracle query and the corresponding protocol transcript is $\mathbf{tr}_\pi = \{\}$, then \mathcal{A}_{prf} returns $a \xleftarrow{\$} \{0, 1\}^\alpha$ and updates $\mathbf{tr}_\pi \leftarrow a$.
- SendReader**($(b, c), \pi$) If π has been previously generated by a **LaunchIdent** oracle query and the corresponding protocol transcript is $\mathbf{tr}_\pi = a$, then \mathcal{A}_{prf} updates the protocol transcript $\mathbf{tr}_\pi \leftarrow (a, b, c)$ and aborts otherwise.
- Result**(π) If π has been previously generated by a **LaunchIdent** oracle query and the corresponding protocol transcript $\mathbf{tr}_\pi = (a, b, c)$ has been generated by $a \leftarrow \text{SendReader}(\emptyset, \pi)$, then \mathcal{A}_{prf} computes $c' \leftarrow F_K(a, b)$ for each (ID, K) in DB. If $c' = c$ for some (ID, K) , \mathcal{A}_{prf} returns 1. If there is no $c' = c$, then \mathcal{A}_{prf} returns 0.
- CorruptTag**($vtag$) If there is a tuple $(\Gamma[vtag], K, S)$ in DB, \mathcal{A}_{prf} returns S . Note that according to Definition 4.4, \mathcal{A}_{sec} is not allowed to corrupt the tag \mathcal{T} and hence, \mathcal{A}_{prf} needs not to simulate the **CorruptTag** oracle for the tag \mathcal{T} .

With non-negligible probability, after a polynomial number of oracle queries, \mathcal{A}_{sec} returns a protocol message (\tilde{b}, \tilde{c}) for a given \tilde{a} . Next, \mathcal{A}_{prf} sends $x \leftarrow (\tilde{a}, \tilde{b})$ to \mathcal{C}^{prf} who responds with a challenge y , which is either $y = F_{\tilde{K}}(x)$ or $y \xleftarrow{\$} \{0, 1\}^\beta$. In case $y = \tilde{c}$, \mathcal{A}_{prf} returns 0 and 1 otherwise.

Note that in case $b = 1$, \mathcal{A}_{prf} perfectly simulates all oracles defined in Section 4.2.2 to \mathcal{A}_{sec} . Hence, in case $b = 1$, by assumption \mathcal{A}_{sec} generates (\tilde{b}, \tilde{c}) for any given \tilde{a} such that $\tilde{c} = F_{\tilde{K}}(\tilde{a}, \tilde{b})$ holds with non-negligible probability. This means that \mathcal{A}_{prf} has a non-negligible advantage of distinguishing the output of F and a random value. Clearly, this contradicts the pseudo-randomness of F (Definition 2.4), which proves Theorem 9.6. \square

Theorem 9.7 (Destructive Privacy). *The RFID protocol illustrated in Figure 9.3 achieves destructive privacy (Definition 4.6) if the protocol achieves tag authentication (Definition 4.4), PUF is an ideal PUF (Definition 9.7) and F is a PRF (Definition 2.4).*

Proof of Theorem 9.7. According to Definition 4.6, destructive privacy means that there is a blinder \mathcal{B} that simulates the **LaunchIdent**, **SendTag**, **SendReader** and **Result** oracle such that no destructive adversary \mathcal{A}_{prv} (Definition 4.3) can distinguish between the blinder \mathcal{B} and the real oracles. Hence, we first give the construction of \mathcal{B} and then show that it cannot be distinguished from the real oracles by any destructive adversary \mathcal{A}_{prv} .

The simulation of the **LaunchIdent** oracle is trivial. \mathcal{B} simulates the **SendTag** and the **SendReader** oracle queries by returning random numbers of the specific output domain. To simulate **Result**, \mathcal{B} returns 1 only if the corresponding protocol transcript has been generated by a **SendReader** and **SendTag** query (i.e., the transcript has been generated by an “honest” tag and reader) and 0 otherwise.

We show by hybrid arguments that, if \mathcal{A}_{prv} can distinguish \mathcal{B} from the real oracles, then we can use \mathcal{A}_{prv} to construct a polynomial time algorithm that violates either tag authentication or the security properties of PUF. Let game $\mathcal{G}^{(0)}$ be the game where \mathcal{A}_{prv} interacts with the real oracles as defined in Section 4.2.2. Then we consider the hybrid game $\mathcal{G}^{(1)}$ that is exactly as $\mathcal{G}^{(0)}$ with the only difference that the states S and the authentication secrets K of all tags are simulated by randomly chosen values. We show that if \mathcal{A}_{prv} can distinguish between $\mathcal{G}^{(0)}$ and $\mathcal{G}^{(1)}$, then we can use \mathcal{A}_{prv} to construct a polynomial time algorithm \mathcal{A}_{puf} that contradicts the security property of PUF (Definition 9.7). The idea is that \mathcal{A}_{puf} uses the PUF-challenger \mathcal{C}^{puf} (cf. Definition 9.7) to simulate the oracles defined in Section 4.2.2 to \mathcal{A}_{prv} . By the contradicting assumption \mathcal{A}_{prv} detects \mathcal{B} with non-negligible probability if the oracle \mathcal{O}^{PUF} provided by \mathcal{C}^{puf} simulates a random function. Hence, the output of \mathcal{A}_{prv} can be used to distinguish between the output of PUF and a random value, which contradicts the security of PUF (Definition 9.7).

Next, we consider the hybrid game $\mathcal{G}^{(2)}$ that is exactly as $\mathcal{G}^{(1)}$ with the only difference that the **SendTag** oracle is simulated by \mathcal{B} as described above. We show that if \mathcal{A}_{prv} can distinguish between $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$, then we can use \mathcal{A}_{prv} to construct a polynomial time algorithm \mathcal{A}_{prf} that contradicts the security property of the PRF F (Definition 2.4). Therefore, \mathcal{A}_{prf} uses the PRF-challenger \mathcal{C}^{prf} to simulate the oracles defined in Section 4.2.2 to \mathcal{A}_{prv} . Since \mathcal{A}_{prv} is assumed to detect \mathcal{B} with non-negligible probability if the oracle \mathcal{O}^{F_K} provided by \mathcal{C}^{prf} simulates a random function, \mathcal{A}_{prf} can use the output of \mathcal{A}_{prv} to distinguish between the output of F_K and a random value with non-negligible probability. Clearly, this violates the security property of the PRF F (Definition 2.4).

We finally consider the hybrid game $\mathcal{G}^{(3)}$ that is exactly as $\mathcal{G}^{(2)}$ with the only difference that the **Result** oracle is simulated by \mathcal{B} as described above. We show that if \mathcal{A}_{prv} can distinguish between $\mathcal{G}^{(2)}$ and $\mathcal{G}^{(3)}$, then \mathcal{A}_{prv} can be used to construct a polynomial time algorithm \mathcal{A}_{sec} that contradicts tag authentication (Definition 4.4). Note that the simulation of **Result** is perfect except for the case where \mathcal{A}_{prf} can generate a protocol transcript (without just forwarding the messages of an uncorrupted honest tag to the

reader) that makes the real **Result** oracle to return 1. However, as shown in the proof of Theorem 9.6 this can only happen with negligible probability. Note that $\mathbf{G}^{(3)}$ corresponds to the game where \mathcal{A}_{prv} interacts with a full blinder \mathcal{B} . Hence, \mathcal{A}_{prv} cannot distinguish between the real oracles and the full blinder \mathcal{B} , which completes the proof of Theorem 9.7.

The details of the proof are as follows: According to Definition 4.6, destructive privacy means that there is a blinder \mathcal{B} that simulates the **LaunchIdent**, **SendTag**, **SendReader** and the **Result** oracle such that no destructive adversary \mathcal{A}_{prv} (Definition 4.3) can distinguish between the blinder \mathcal{B} and the real oracles. Hence, to prove Theorem 9.7, we first give the construction of the blinder \mathcal{B} and then show that it cannot be distinguished from the real oracles by any destructive adversary \mathcal{A}_{prv} .

The blinder \mathcal{B} is initialized with the security parameters $l, \gamma, \kappa, \alpha, \beta$ and the public key $pk_{\mathcal{R}}$ of the reader \mathcal{R} and works as follows:

LaunchIdent() The simulation of the **LaunchIdent** oracle is straightforward.

SendTag($a, vtag$) Return $b \xleftarrow{\$} \{0, 1\}^\alpha$ and $c \xleftarrow{\$} \{0, 1\}^\beta$.

SendReader(π) Return $a \xleftarrow{\$} \{0, 1\}^\alpha$.

SendReader($(b, c), \pi$) Since oracle queries of this form do not generate any output nor change the state of the tag and the reader, the blinder \mathcal{B} needs not to simulate their responses.

Result(π) If π has been previously generated by a **LaunchIdent** oracle query and the corresponding protocol transcript $\text{tr}_\pi = (a, b, c)$ has been obtained through $a \leftarrow \text{SendReader}(\emptyset, \pi)$ and $(b, c) \leftarrow \text{SendTag}(a, vtag)$, return 1 and 0 otherwise.

In the following, we show that, if there is a destructive adversary \mathcal{A}_{prv} who can distinguish the blinder \mathcal{B} from the real oracles, then we can use \mathcal{A}_{prv} to construct a polynomial time algorithm that violates either tag authentication (Definition 4.4) or the security properties of the underlying PRF F (Definition 2.4) or PUF (Definition 9.7).

Let game $\mathbf{G}^{(0)}$ be the game where the adversary \mathcal{A}_{prv} interacts with the real oracles as defined in Section 4.2.2. Now consider the following hybrid game $\mathbf{G}^{(1)}$ that is exactly as $\mathbf{G}^{(0)}$ with the only difference that the states S and the authentication secrets K of all tags are simulated by randomly chosen values. In the following, we show that if \mathcal{A}_{prv} can distinguish between $\mathbf{G}^{(0)}$ and $\mathbf{G}^{(1)}$, then we can use \mathcal{A}_{prv} to construct a polynomial time algorithm \mathcal{A}_{puf} that contradicts the security property of PUF (Definition 9.7).

According to the protocol specification given in Section 9.3.1, the states and PUFs of different tags are chosen independently. Moreover, \mathcal{A}_{puf} can trivially simulate different tags by following the protocol specifications. Hence, we assume w.l.o.g. that \mathcal{A}_{prv} creates just one single tag \mathcal{T} during his attack. To create \mathcal{T} , \mathcal{A}_{puf} chooses $S \xleftarrow{\$} \{0, 1\}^\gamma$ and sets $K \leftarrow \mathcal{O}^{\text{PUF}}(S)$. Note that $\mathcal{O}^{\text{PUF}}(S)$ either returns $K \leftarrow \text{PUF}(S)$ as in $\mathbf{G}^{(0)}$ or $K \xleftarrow{\$} \{0, 1\}^\kappa$ as in $\mathbf{G}^{(1)}$. Now, \mathcal{A}_{puf} can interact with all the oracles defined in Section 4.2.2 that are simulated by \mathcal{A}_{puf} based on the input of \mathcal{C}^{puf} . The simulation of the **DrawTag**, **FreeTag** and **LaunchIdent** oracle is straightforward. Note that the output of the **Result** and **CorruptTag** oracle is independent of the PUF of \mathcal{T} and hence, these oracles can be simulated in a trivial way. Since **SendReader** queries generate no output and do not change the state S of \mathcal{T} , they need not be simulated by \mathcal{A}_{puf} . On a **SendTag**($a, vtag$) oracle query, \mathcal{A}_{puf} responds with $b \xleftarrow{\$} \{0, 1\}^\alpha$ and $c \leftarrow F_K(a, b)$.

Note that \mathcal{A}_{prv} is a destructive adversary and hence, by making a **CorruptTag**($vtag$) query, \mathcal{A}_{prv} can obtain the state S of $vtag$ but he can no longer send any query that involves the tag $vtag$ afterwards. After a polynomial number of oracle queries, \mathcal{A}_{prv} returns a bit b' . In case $b' = 1$ (which indicates that \mathcal{A}_{prv} detected \mathcal{B}), with non-negligible probability \mathcal{O}^{PUF} must have returned a random $K \xleftarrow{\$} \{0, 1\}^\kappa$. Hence, \mathcal{A}_{puf} can distinguish between the output of a PUF and a randomly chosen value, which contradicts the security property of the PUF (Definition 9.7). As a result, the following is negligible:

$$|\Pr[\mathbf{G}^{(0)} = 1] - \Pr[\mathbf{G}^{(1)} = 1]|. \quad (9.1)$$

Next, consider the hybrid game $\mathbf{G}^{(2)}$ that is exactly as $\mathbf{G}^{(1)}$ with the only difference that the **SendTag** oracle is simulated by the blinder \mathcal{B} as described above. In the following, we show that if \mathcal{A}_{prv} can distinguish between $\mathbf{G}^{(1)}$ and $\mathbf{G}^{(2)}$, then we can use \mathcal{A}_{prv} to construct a polynomial time algorithm \mathcal{A}_{prf} that contradicts the security property of the PRF F (Definition 2.4).

Let $q \in \mathbb{N}$ be the number of **SendTag** queries made by \mathcal{A}_{prv} , which is polynomial in the security parameter l . Moreover, let $i \in \{0, \dots, q\}$. Now consider the following hybrid game \mathbf{G}_i with \mathcal{A}_{prv} : The first i **SendTag** queries of \mathcal{A}_{prv} are answered by the blinder \mathcal{B} (as in $\mathbf{G}^{(2)}$), while the remaining $q - i$ queries are forwarded and answered by the real **SendTag** oracle (as in $\mathbf{G}^{(1)}$). Note that \mathbf{G}_0 corresponds to $\mathbf{G}^{(1)}$ whereas \mathbf{G}_q corresponds to game $\mathbf{G}^{(2)}$. Hence and due to the contradicting assumption made at the beginning of the proof, it holds that $\mathbf{Adv}_{\mathcal{A}_{\text{prv}}}^{\text{prv}} = |\Pr[\mathbf{G}_0 = 1] - \Pr[\mathbf{G}_q = 1]|$ is non-negligible. Therefore,

there must be some index $i \in \{1, \dots, q\}$ such that

$$|\Pr[\mathbf{G}_{i-1} = 1] - \Pr[\mathbf{G}_i = 1]| \quad (9.2)$$

is non-negligible. Note that Equation 9.2 implies w.l.o.g. that \mathcal{A}_{prv} detects \mathcal{B} in game \mathbf{G}_i with non-negligible probability while \mathcal{A}_{prv} can detect \mathcal{B} in game \mathbf{G}_{i-1} only with negligible probability.

We can use \mathcal{A}_{prv} to construct the following polynomial time algorithm \mathcal{A}_{prf} that violates the security property of the PRF F (Definition 2.4). Therefore, \mathcal{A}_{prf} plays the hybrid game \mathbf{G}'_i with \mathcal{A}_{prv} , which is like \mathbf{G}_i except that the i -th $\text{SendTag}(a, vtag)$ query is answered as follows: \mathcal{A}_{prf} chooses $b \xleftarrow{\$} \{0, 1\}^\alpha$ and sends $x \leftarrow (a, b)$ to the PRF-challenger \mathcal{C}^{prf} , which responds with $y \leftarrow \mathcal{O}^F(x)$ that is either $y = F_K(x)$ or $y \xleftarrow{\$} \{0, 1\}^{2\alpha}$. Then, \mathcal{A}_{prf} sends (b, c) to \mathcal{A}_{prv} . Note that, in case \mathcal{C}^{prf} sends $y = F_K(x)$ then $\mathbf{G}'_i = \mathbf{G}_{i-1}$ and $\mathbf{G}'_i = \mathbf{G}_i$ otherwise. Hence, if \mathcal{A}_{prv} returns 1 (which indicates that \mathcal{A}_{prv} detected \mathcal{B}) then \mathcal{A}_{prf} must have played \mathbf{G}_i . Clearly, this allows \mathcal{A}_{prf} to distinguish the output of the PRF F from a random value, which contradicts the security property of the PRF (Definition 2.4). Hence, the PRF ensures that Equation 9.2 is negligible and thus, the following is negligible:

$$|\Pr[\mathbf{G}^{(1)} = 1] - \Pr[\mathbf{G}^{(2)} = 1]|. \quad (9.3)$$

Next, consider the hybrid game $\mathbf{G}^{(3)}$ that is exactly as $\mathbf{G}^{(2)}$ with the only difference that the **Result** oracle is simulated by the blinder \mathcal{B} as described above. In the following, we show that if there is an adversary \mathcal{A}_{prv} who can distinguish between $\mathbf{G}^{(2)}$ and $\mathbf{G}^{(3)}$, then \mathcal{A}_{prv} can be used to construct a polynomial time algorithm \mathcal{A}_{sec} that contradicts tag authentication (Definition 4.4).

In the following, let $p \in \mathbb{N}$ be the number of **Result** queries made by \mathcal{A}_{prv} , which is polynomial in the security parameter l . Moreover, let $i \in \{0, \dots, p\}$. Now consider the following hybrid game \mathbf{G}^*_i : The first i **Result** queries of \mathcal{A}_{prv} are answered by the blinder \mathcal{B} (as in $\mathbf{G}^{(3)}$), while the remaining $p - i$ queries are forwarded and answered by the real **Result** oracle (as in $\mathbf{G}^{(2)}$). Note that \mathbf{G}^*_0 corresponds to $\mathbf{G}^{(2)}$ whereas \mathbf{G}^*_p is equivalent to $\mathbf{G}^{(3)}$. Hence and due to the contradicting assumption made at the beginning of the proof, it holds that $\mathbf{Adv}_{\mathcal{A}_{\text{prv}}}^{\text{prv}} = |\Pr[\mathbf{G}^*_0 = 1] - \Pr[\mathbf{G}^*_p = 1]|$ is non-negligible. Therefore, there

must be some index $i \in \{1, \dots, p\}$ such that

$$|\Pr[\mathbf{G}_{i-1}^* = 1] - \Pr[\mathbf{G}_i^* = 1]| \quad (9.4)$$

is non-negligible. Note that Equation 9.4 implies that w.l.o.g. \mathcal{A}_{prv} detects \mathcal{B} in game \mathbf{G}_i^* with non-negligible probability while he has at most negligible probability to detect \mathcal{B} in game \mathbf{G}_{i-1}^* . This means that in \mathbf{G}_i^* \mathcal{A}_{prv} runs a protocol instance π where the **Result** oracle simulated by \mathcal{B} returns a different output than the real **Result** oracle. According to the description of \mathcal{B} given at the beginning of this proof and the definition of the **Result** oracle in Section 4.2.2, this can only happen if \mathcal{A}_{prv} generates a protocol transcript $\text{tr}_\pi = (a, b, c)$ such that $c = F_K(a, b)$ where $(ID, K) \in \text{DB}$ and tag \mathcal{T} has not been corrupted by \mathcal{A}_{prv} . However, as shown in the proof of Theorem 9.6 this can only happen with negligible probability. Hence, tag authentication ensures that Equation 9.5 is negligible and thus the following is negligible as well:

$$|\Pr[\mathbf{G}^{(2)} = 1] - \Pr[\mathbf{G}^{(3)} = 1]|. \quad (9.5)$$

Note that $\mathbf{G}^{(3)}$ corresponds to the game where \mathcal{A}_{prv} interacts with a full blinder \mathcal{B} . Hence, from Equations 9.1, 9.3 and 9.5 it follows that $|\Pr[\mathbf{G}^{(0)} = 1] - \Pr[\mathbf{G}^{(3)} = 1]|$ is negligible. This means that \mathcal{A}_{prv} cannot distinguish between the real oracles and the full blinder \mathcal{B} , which completes the proof of Theorem 9.7. \square

9.3.3 Conclusion

We have shown that, while PUFs allow to uniquely identify devices, they can also be used to enhance the privacy in RFID systems. Specifically, we showed a privacy-preserving tag authentication protocol for RFID that can be proven to be destructive private in a variant of the V-Model. The protocol is based on the weak private protocol proposed by Vaudenay [190] and uses Physically Unclonable Functions (PUFs) as tamper-evident key storage in a similar way as described by Tuyls et al. [184]. The properties of the PUF ensure that any attempt to physically tamper with the PUF to obtain the authentication secret of the tag result in the destruction of the PUF and the tag secret, which corresponds to the definition of a destructive adversary in the V-Model.

10 Application Example: RFID-based E-Tickets

Electronic tickets for public transportation is one of many practical RFID-based applications that is already widely deployed in practice and will become more popular in the future [141, 178, 177]. However, RFID e-ticket systems currently used in practice usually do not consider privacy aspects (i.e., the confidentiality of the identity and location of users). Moreover, e-tickets must fulfill strict usability requirements in order to be competitive to conventional paper-based tickets.

10.1 General Scenario

An e-ticket system, as shown in Figure 10.1, consists of at least one ticket issuing entity (*issuer*), a set of *users*, *tickets* and *readers* that verify whether tickets are valid. Since we are focusing on RFID-based systems where tickets are realized as RFID tags, in the following we use *ticket* synonymously to *tag*. Typically, a user \mathcal{U} must obtain a ticket \mathcal{T} from an issuer \mathcal{I} . Therefore, user \mathcal{U} selects his desired ticket. Issuer \mathcal{I} then checks whether user \mathcal{U} is eligible to obtain that ticket (e.g., whether \mathcal{U} paid for it) and, if applicable, issues the ticket \mathcal{T} and passes it to \mathcal{U} . From now on, \mathcal{U} can use \mathcal{T} to prove that he is authorized to use the transit network. This means that every user who is in possession of a ticket that has been issued by a genuine issuer is considered to be an *authorized user*.

Now assume that, as shown in Figure 10.1, user \mathcal{U} wants to travel from a place X to some location Y . Before \mathcal{U} is allowed to enter the transit system at X , he must first prove to a reader \mathcal{R}_{in} at the entrance of the transit network that he is authorized to access it. If reader \mathcal{R}_{in} can successfully verify the user's ticket \mathcal{T} , \mathcal{U} is allowed to enter. Otherwise access is denied. During his trip, \mathcal{U} may encounter arbitrary inspections where he must prove that he is authorized to use the transit network. Thus, a reader \mathcal{R} may check the user's ticket \mathcal{T} . If the verification of \mathcal{T} is successful, \mathcal{U} is allowed to continue his trip. Otherwise, \mathcal{U} must leave the transit network and may be punished for using it without authorization. After arriving at Y , the user's ticket \mathcal{T} may be checked for a last time. Again, if \mathcal{T} cannot be verified successfully, \mathcal{U} may be punished.

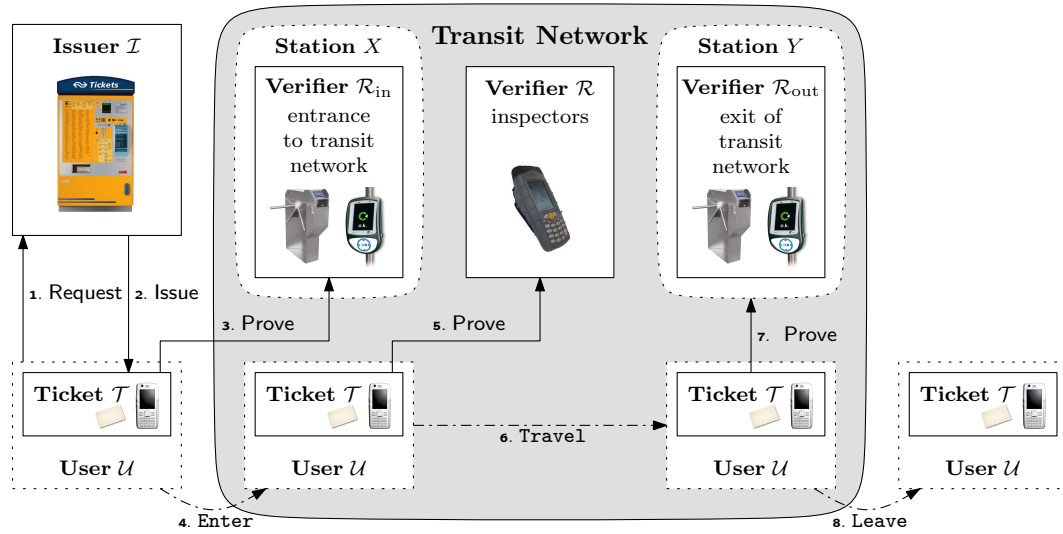


Figure 10.1: General Scenario of E-Tickets.

Note that authentication is typically bound to some limitations. For instance, this may be some geographical or timely usage restriction that must also be considered during ticket verification.

Remark. Parts of this chapter have been published in [163] and [166].

10.2 Requirement Analysis

An RFID-based e-ticket system should fulfill the following requirements. An in-depth discussion of these requirements is provided in Section 3.1.4.

Security Goals

- *Authentication*: It is infeasible for the adversary to make an honest reader accept.
- *Unclonability*: It is infeasible for the adversary to duplicate a valid tag.
- *Availability*: It is infeasible for the adversary to alter tags by interacting with them over the wireless interface.

Privacy Goals

- *Confidentiality*: It should be infeasible for the adversary to access to user data.
- *Anonymity*: It should be infeasible for the adversary to identify tags.
- *Untraceability*: It should be infeasible for an adversary with access to the current state of a tag \mathcal{T} to (1) trace \mathcal{T} in any *previously recorded* protocol run and (2) to trace any tag different from \mathcal{T} .

Note that this notion of untraceability is an extension of *forward privacy* (cf. Section 4.2.4) and sufficient for most practical applications [152]. Untraceability implies anonymity and confidentiality since an adversary who can access user-specific data on tags and/or identify tags can also trace them. Further, as discussed in Section 3.1.4, an RFID system cannot protect against powerful adversaries who can trace users, e.g., by visually observing them.

Functional Goals

- *Efficiency*: Verification of tags must be fast.
- *Scalability*: A large number of tags must be supported.

10.3 E-Ticket Systems in Practice

Most e-ticket systems in practice are proprietary solutions whose specifications are not publicly available. This section reviews the most common approach of implementing authentication of e-tickets in practice by the example of the Calypso e-ticket system [178], of which at least some information is public. Moreover, to the best of our knowledge, there is no solution for RFID-based e-tickets in practice that explicitly considers the location privacy of users.

10.3.1 Calypso E-Ticket Standard

Calypso is an e-ticket standard based on RFID that is widely used in Europe and North and South America [178]. The roles in the Calypso system correspond to the model presented in Figure 10.1. However, Calypso does not consider the privacy of users and thus does not fulfill any of the privacy requirements in Section 10.2. Actually, all transactions

involving a Calypso e-ticket provide no confidentiality at all [178]. Moreover, Calypso tickets can store personal data of their owner (e.g., his name) that can be queried by every verifier. Thus the Calypso e-ticket system leaks user-related information and allows the creation of location profiles by everyone who is in possession of a standard RFID reader. All messages of a Calypso ticket are authenticated by a symmetric-key-based authentication mechanism. Calypso seems to fulfill all of the security requirements but none of the privacy requirements in Section 10.2.

Calypso implements a common approach to authenticate low-cost RFID tags based on a simple challenge-response protocol. Each tag has a symmetric authentication key K that can be computed as a function of the serial number s of the tag and a global master secret. All readers are equipped with a tamper-resistant security module that knows and protects this master secret and can be used as a black-box to compute K from s . To authenticate a tag, a reader sends a random challenge c to the tag, which then computes $h \leftarrow f(K, c)$ where f is some pseudo-random function. Finally, the tag returns (s, h) to the reader that uses its security module to derive K and then verifies h . In case the verification is successful, the tag has been authenticated. Obviously, this approach cannot provide location privacy since all transactions of a tag can be linked by its serial number s that is transmitted in clear in every protocol run. All subsequent transactions to update or to read data from a Calypso ticket are authenticated this way but they are not encrypted.

10.3.2 Other E-Ticket Systems

There are several other proprietary solutions for e-tickets in practice. Most of them are based on widely used RFID tags. Prominent examples are FeliCa [177] and MiFare [141]. FeliCa [177] is a contactless smartcard by Sony that is mainly used in the Asia-Pacific area for different purposes, including e-tickets for public transportation. MiFare is a family of contactless smartcards produced by Philips/NXP Semiconductors. These tags are widely used for different purposes, including e-tickets for public transportation. There were several publications on attacks against MiFare Classic tags [140, 64], that use a proprietary encryption algorithm which has been completely broken [43]. However, other MiFare products are claimed not to be affected.

The attacks on MiFare Classic tags demonstrate a major problem of proprietary security solutions: Manufacturers of low-cost hardware try to find a compromise between effi-

ciency and security of their products. Thus, they often implement proprietary lightweight cryptographic algorithms which of the specifications are not public and thus are typically not sufficiently evaluated. As for MiFare Classic, these algorithms can often be reverse-engineered, which allows cryptanalysis or efficient key search by running the algorithms on more powerful hardware. In the case of MiFare Classic both ways allowed to break the security goals of these tags at a point in time where they were already widely used in practice.

10.4 Secure and Privacy-preserving Protocols for E-Tickets

10.4.1 Existing Solutions

Privacy-preserving e-tickets are discussed in a few papers. In [86], the authors sketch an anonymous payment system for public transit based on anonymous credentials [39] and e-cash [37]. They propose tickets to be managed either by RFID tags or mobile computing devices like mobile phones or PDAs. As pointed out in Section 3.3.2, anonymous credentials and e-cash are not applicable to currently available RFID devices whereas the use of mobile phones or PDAs for managing e-tickets introduces several other drawbacks.

10.4.2 Anonymizer-based Solutions

Anonymizer-enabled protocols provide an easy and cost-efficient way to allow operators of RFID systems to enable privacy for the concerned users of ticket systems (who may buy his/her own personal anonymizer) with only minor extra costs. The main advantage of this approach is that existing RFID technology can be used without requiring additional hardware on the ticket. Anonymizers can be realized as a software on the users' NFC-enabled smartphones or integrated into the RFID readers used to verify the tickets. Depending on the concrete requirements on the ticket system either of the two schemes described in Chapter 5 can be used to perform privacy-preserving authentications of tickets. While the protocols described in Section 5.3.2 are highly efficient in the sense that they require the tags to perform only a lightweight cryptographic hash function and some basic arithmetic operations (such as addition or multiplication), the scheme in Section 5.4.2 achieves full anonymity even against a collusion of malicious verifiers and anonymizers at the cost of higher computational and storage requirements on the tag.

10.4.3 PUF-based Solutions

Physically Unclonable Functions (PUFs) are a very promising approach to enhance the cloning-resistance of cost-efficient RFID-based electronic tickets. As discussed in Chapter 9, PUFs and the required post-processing algorithms (error correction and privacy-amplification) can be implemented with minimal costs. The PUF-based authentication scheme presented in Section 9.2.2 can be combined with different PUF types and lightweight hash algorithms that allow for highly area and power optimized implementations that are perfectly suited for the authentication of RFID-based electronic tickets. However, the helper data transferred from the tag to the reader typically leaks some information on the underlying PUF response that may be sufficient to identify and trace the ticket.

One approach that benefits from the cloning-resistance of PUFs and at the same time uses the PUF to enhance user privacy is described in Section 9.3.1. This protocol achieves secure tag authentication and one of the strongest privacy notions, destructive privacy, in a variant of the V-Model. This means that the protocol provides untraceability of tickets against adversaries that permanently destroy the ticket by physically attacking it, e.g., in an attempt to reverse-engineer the ticket hardware.

11 Conclusion

11.1 Summary

In this work, we addressed the problem of designing efficient, secure and privacy-preserving authentication protocols that are applicable to real-world RFID systems. Specifically, we pointed out subtle issues in state-of-the art RFID security and privacy models, proposed the first security framework for anonymizer-enabled RFID systems and designed novel anonymizer-based privacy-preserving authentication schemes. We advanced the existing work on PUFs and PUF-based authentication by analyzing the most common electronic PUF types implemented in ASIC, introducing a formal security framework for PUFs and presenting practical lightweight and privacy-preserving PUF-based authentication protocols.

Analysis of the state-of-the-art in RFID security and privacy. We analyzed one of the most comprehensive RFID security and privacy models [190, 150], which generalizes and improves many previous works. We pointed out weaknesses and deficiencies in this model and investigated some subtle issues such as tag corruption aspects. More detailed, we showed that the formal definition of tag corruption discloses the temporary memory content of tags and leads to the impossibility of achieving both mutual authentication and any reasonable notion of privacy in their model. Moreover, we showed that the strongest privacy notion (*narrow-strong privacy*) cannot be achieved simultaneously with reader authentication even under the strong assumption that tag corruption does not disclose the temporary memory content of tags. These results led to the refinement of the Paise and Vaudenay RFID security and privacy model [191] and they were considered in several subsequent works on RFID systems [84, 53].

Security & privacy framework for anonymizer-enabled RFID systems. We presented the first security and privacy framework for anonymizer-enabled RFID systems and two privacy-preserving RFID authentication schemes using anonymizers. Both schemes achieve several appealing features that were not simultaneously achieved by any

previous proposal. The first scheme is very efficient for all involved entities, in particular for the tags that only have to perform minimal computations. The protocol achieves privacy under tag corruption and is secure against impersonation attacks and forgeries even if the adversary can corrupt the anonymizers. The second scheme provides anonymity and untraceability of tags against readers as well as secure tag authentication against collisions of malicious readers and anonymizers using cost-efficient tags that cannot perform public-key cryptography (i.e., modular exponentiations).

First large-scale analysis of PUF implementations in ASIC. A promising approach to enhance the cloning-resistance of RFID tags with minimal overhead on the tag side are Physically Unclonable Functions (PUFs). We presented the first large-scale security analysis of ASIC implementations of the five most popular electronic PUF types, including Arbiter, Ring Oscillator, SRAM, Flip-flop and Latch PUFs that are suitable for the integration into RFID tags. Our analysis is based on PUF data obtained at different operating conditions from 96 ASICs containing multiple PUF instances, which have been manufactured in TSMC 65 nm CMOS technology. We presented an evaluation methodology and quantified the robustness and unpredictability properties of PUF responses, which are fundamental for the integration of PUFs into cryptographic primitives and protocols, such as authentication schemes. Since all PUFs have been implemented in the same ASIC and analyzed with the same evaluation methodology, our results allow for the first time a fair comparison of their properties.

Our evaluation results show that all PUFs in the ASIC are sufficiently robust for practical applications. However, not all of them achieve the unpredictability property. In particular, the responses of Arbiter PUFs have very low entropy, while the entropy of Flip-flop and Latch PUF responses is affected by temperature variations. In contrast, the Ring Oscillator and SRAM PUFs seem to achieve all desired properties of a PUF: Their challenge/response behavior hardly changes under different operating conditions and the entropy of their responses is quite high. Furthermore, the responses generated by different Ring Oscillator and SRAM PUF instances seem to be independent, which means that the adversary cannot predict the response of a PUF based on the challenge/response pairs of another PUF. However, the min-entropy, i.e., the minimum number of random bits observed in a response of the Ring Oscillator PUF, is low, which means that some responses can be guessed with high probability.

Formal security framework for PUFs. We present a formal foundation for security primitives based on PUFs, focussing on the main properties at the heart of most published works on PUFs: robustness, unclonability and unpredictability. Our work allows for a meaningful security analysis of security primitives taking advantage of physical properties, becoming increasingly important in the development of the next generation of secure information systems. Since its publication, our framework has been used to estimate the robustness and unclonability properties of image-based PUFs [174], in the context of designing anti-counterfeiting mechanisms [173] and physical hash functions [58].

Efficient PUF-enhanced RFID security and privacy. We presented two PUF-based authentication schemes that overcome the practical and security problems of existing approaches. In contrast to existing PUF-based authentication schemes, the first protocol supports PUF-based mutual authentication between tags and readers, is resistant to emulation attacks against the underlying PUF and highly scalable since (in contrast to most existing approaches) it does not require the reader to store a large number of PUF challenge/response pairs. The scheme is based on reverse fuzzy extractors [85], a new approach to correct noise in PUF responses that allows for extremely lightweight implementations on the tag. Further, it supports Logically Reconfigurable PUFs (LR-PUFs) [108, 109] that enable secure updates of the tag authentication secrets bound to the PUF by changing the PUF-behaviour after the deployment of the tag. The second protocol uses the PUF as a secure key storage and addresses an open question on the feasibility of destructive privacy [190], i.e., the privacy of tags that are destroyed during tag corruption.

11.2 Directions for Future Research

Practical PUF designs. Many known electronic PUFs can be compromised: Memory-based PUFs can be read out completely since they have only a limited response space and most delay-based PUFs can be emulated using machine learning techniques (cf. Section 6.5.1). While these PUFs can still be used in many applications, such as PUF-based key storage (cf. Section 6.4.2) and Controlled PUFs (cf. Section 6.6.1), where the adversary cannot access the challenge/response pairs of the PUF, the use of these PUFs in applications with strong unclonability and unpredictability requirements, such as device authentication schemes (cf. Section 6.4.1 and Chapter 9), must be carefully considered.

Furthermore, PUF responses can be verified only when the verifier has access to a database of previously recorded challenge/response pairs (CRPs), which may lead to scalability problems in practice. Hence, one open challenge is the development and implementation of novel PUF designs that achieve the requirements of many existing theoretical PUF-based security solutions in the literature, including resistance to emulation attacks, large (ideally exponential) challenge/response space to prevent complete readout of the PUF, public verifiability (i.e., no CRP database required to verify the PUF response), tamper-evidence, physical reconfigurability and a small hardware footprint.

Side-channel analysis of PUFs. Many PUF-based applications, such as PUF-based key storage (cf. Section 6.4.2), require PUF responses to be inaccessible to the adversary, which is typically justified by the assumption of the PUF being tamper-evident so that any attempt to physically access the PUF response (such as an invasive attack) permanently changes the challenge/response behavior of the PUF. However, even when a tamper-evident PUF (such as a Coating PUF) is used, it is currently unclear whether existing PUF implementations in integrated circuits leak information on their response over side channels, such as electro-magnetic radiation, power consumption or remanence decay effects. Hence, the analysis of the side-channel leakage of known PUF implementations is an interesting open research problem.

Bibliography

- [1] ECRYPT II yearly report on algorithms and key sizes (2010-2011).
<http://www.ecrypt.eu.org/documents/D.SPA.17.pdf>, June 2011.
- [2] Frederik Armknecht, Liqun Chen, Ahmad-Reza Sadeghi, and Christian Wachsmann. Anonymous authentication for RFID systems. In Siddika Ors Yalcin, editor, *Radio Frequency Identification: Security and Privacy Issues (RFIDSec)*, volume 6370 of *Lecture Notes in Computer Science (LNCS)*, pages 158–175. Springer, Berlin/Heidelberg, Germany, June 2010.
- [3] Frederik Armknecht, Roel Maes, Ahmad-Reza Sadeghi, François-Xavier Standaert, and Christian Wachsmann. A formal foundation for the security features of physical functions. In *IEEE Symposium on Security and Privacy (S&P)*, pages 397–412. IEEE, Washington, DC, USA, May 2011.
- [4] Frederik Armknecht, Roel Maes, Ahmad-Reza Sadeghi, Berk Sunar, and Pim Tuyls. Memory leakage-resilient encryption based on physically unclonable functions. In Mitsuru Matsui, editor, *Advances in Cryptology (ASIACRYPT)*, volume 5912 of *Lecture Notes in Computer Science (LNCS)*, pages 685–702. Springer, Berlin/Heidelberg, Germany, December 2009.
- [5] Frederik Armknecht, Ahmad-Reza Sadeghi, Alessandra Scafuro, Ivan Visconti, and Christian Wachsmann. Impossibility results for RFID privacy. In Marina Gavrilova, C. J. Kenneth Tan, and Edward Moreno, editors, *Transactions on Computational Science XI*, volume 6480 of *Lecture Notes in Computer Science (LNCS)*, pages 39–63. Springer, Berlin/Heidelberg, Germany, December 2010.
- [6] Frederik Armknecht, Ahmad-Reza Sadeghi, Ivan Visconti, and Christian Wachsmann. On RFID privacy with mutual authentication and tag corruption. In Jianying Zhou and Moti Yung, editors, *Applied Cryptography and Network Security (ACNS)*, volume 6123 of *Lecture Notes in Computer Science (LNCS)*, pages 493–510. Springer, Berlin/Heidelberg, Germany, June 2010.
- [7] Giuseppe Ateniese, Jan Camenisch, and Breno de Medeiros. Untraceable RFID tags via insubvertible encryption. In *ACM Conference on Computer and Communications Security (CCS)*, pages 92–101. ACM, New York, NY, USA, November 2005.
- [8] Atmel Corporation. Atmel CryptoRF.
<http://www.atmel.com/products/other/securerf/>, January 2013.
- [9] Atmel Corporation. RF identification products.
<http://www.atmel.com/products/wireless/rfid/>, January 2013.
- [10] Gildas Avoine. Adversarial model for radio frequency identification. Cryptology ePrint Archive, Report 2005/049, February 2005.

- [11] Gildas Avoine, Etienne Dysli, and Philippe Oechslin. Reducing time complexity in RFID systems. In Bart Preneel and Stafford Tavares, editors, *Selected Areas in Cryptography (SAC)*, volume 3897 of *Lecture Notes in Computer Science (LNCS)*, pages 291–306. Springer, Berlin/Heidelberg, Germany, March 2006.
- [12] Gildas Avoine, Cédric Lauradoux, and Tania Martin. When compromised readers meet RFID. In Heung Y. Youm and Moti Yung, editors, *International Workshop on Information Security Applications (WISA)*, volume 5932 of *Lecture Notes in Computer Science (LNCS)*, pages 36–50. Springer, Berlin/Heidelberg, Germany, August 2009.
- [13] Gildas Avoine and Aslan Tchamkerten. An efficient distance bounding RFID authentication protocol: Balancing false-acceptance rate and memory requirement. In Pierangela Samarati, Moti Yung, Fabio Martinelli, and ClaudioA Ardagna, editors, *Information Security Conference (ISC)*, volume 5735 of *Lecture Notes in Computer Science (LNCS)*, pages 250–261. Springer, Berlin/Heidelberg, Germany, September 2009.
- [14] Lejla Batina, Jorge Guajardo, Bart Preneel, Pim Tuyls, and Ingrid Verbauwhede. Public-key cryptography for RFID tags and applications. In Paris Kitsos and Yan Zhang, editors, *RFID Security*, pages 317–348. Springer US, August 2009.
- [15] Mihir Bellare, Anand Desai, Eron Jorikpi, and Philip Rogaway. A concrete security treatment of symmetric encryption. In *Annual Symposium on Foundations of Computer Science (FOCS)*, pages 394–403. IEEE, Washington, DC, USA, October 1997.
- [16] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In Hugo Krawczyk, editor, *Advances in Cryptology (CRYPTO)*, volume 1462 of *Lecture Notes in Computer Science (LNCS)*, pages 26–45. Springer, Berlin/Heidelberg, Germany, August 1998.
- [17] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM Conference on Computer and Communications Security (CCS)*, pages 62–73. ACM, New York, NY, USA, November 1993.
- [18] Thomas Beth and Yvo Desmedt. Identification tokens — Or: Solving the chess grandmaster problem. In Alfred J. Menezes and Scott A. Vanstone, editors, *Advances in Cryptology (CRYPTO)*, volume 537 of *Lecture Notes in Computer Science (LNCS)*, pages 169–176. Springer, Berlin/Heidelberg, August 1991.
- [19] Mudit Bhargava, Cagla Cakir, and Ken Mai. Comparison of bi-stable and delay-based physical unclonable functions from measurements in 65nm bulk CMOS. In *Custom Integrated Circuits Conference (CICC)*, pages 1–4. IEEE, Washington, DC, USA, September 2012.
- [20] Patrik Bichsel, Jan Camenisch, Thomas Gross, and Victor Shoup. Anonymous credentials on a standard Java card. In *ACM Conference on Computer and Communications Security (CCS)*, pages 600–610. ACM, New York, NY, USA, November 2009.
- [21] Erik O. Blass, Anil Kurmus, Refik Molva, and Thorsten Strufe. PSP: Private and secure payment with RFID. In *ACM Workshop on Privacy in the Electronic Society (WEPS)*, pages 51–60. ACM, New York, NY, USA, October 2009.

-
- [22] Andrey Bogdanov, Miroslav Knežević, Gregor Leander, Deniz Toz, Kerem Varıcı, and Ingrid Verbauwhede. SPONGENT: A lightweight hash function. In Bart Preneel and Tsuyoshi Takagi, editors, *Cryptographic Hardware and Embedded Systems (CHES)*, volume 6917 of *Lecture Notes in Computer Science (LNCS)*, pages 312–325. Springer, April 2011.
 - [23] Jens M. Bohli and Andreas Pashalidis. Relations among privacy notions. *ACM Transactions on Information and System Security*, 14(1), June 2011.
 - [24] Leonid Bolotnyy and Gabriel Robins. Physically unclonable function-based security and privacy in RFID systems. In *Conference on Pervasive Computing and Communications (PerCom)*, pages 211–220. IEEE, Washington, DC, USA, March 2007.
 - [25] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the Weil pairing. In Colin Boyd, editor, *Advances in Cryptology (ASIACRYPT)*, volume 2248 of *Lecture Notes in Computer Science (LNCS)*, pages 514–532. Springer, Berlin/Heidelberg, Germany, December 2001.
 - [26] Christoph Bösch, Jorge Guajardo, Ahmad-Reza Sadeghi, Jamshid Shokrollahi, and Pim Tuyls. Efficient helper data key extractor on FPGAs. In Elisabeth Oswald and Pankaj Rohatgi, editors, *Cryptographic Hardware and Embedded Systems (CHES)*, volume 5154 of *Lecture Notes in Computer Science (LNCS)*, pages 181–197. Springer, Berlin/Heidelberg, Germany, July 2008.
 - [27] Xavier Boyen. Reusable cryptographic fuzzy extractors. In *ACM Conference on Computer and Communications Security (CCS)*, pages 82–91. ACM, New York, NY, USA, October 2004.
 - [28] Stefan Brands and David Chaum. Distance-Bounding protocols. In Tor Helleseth, editor, *Advances in Cryptology (EUROCRYPT)*, volume 765 of *Lecture Notes in Computer Science (LNCS)*, pages 344–359. Springer, Berlin/Heidelberg, Germany, May 1994.
 - [29] Julien Bringer and Hervé Chabanne. Trusted-HB: A low-cost version of HB secure against man-in-the-middle attacks. *IEEE Transactions on Information Theory*, 54(9):4339–4342, September 2008.
 - [30] Julien Bringer, Hervé Chabanne, and Emmanuelle Dottax. HB^{++} : A lightweight authentication protocol secure against some attacks. In *Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing (SecPerU)*, pages 28–33. IEEE, Washington, DC, USA, June 2006.
 - [31] Julien Bringer, Hervé Chabanne, and Thomas Icart. Improved privacy of the tree-based hash protocols using physically unclonable function. In Rafail Ostrovsky, Roberto De Prisco, and Ivan Visconti, editors, *Security and Cryptography for Networks (SCN)*, volume 5229 of *Lecture Notes in Computer Science (LNCS)*, pages 77–91. Springer, Berlin/Heidelberg, Germany, August 2008.
 - [32] Julien Bringer, Hervé Chabanne, and Thomas Icart. Efficient zero-knowledge identification schemes which respect privacy. In *ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, pages 195–205. ACM, New York, NY, USA, March 2009.

- [33] Christina Brzuska, Marc Fischlin, Heike Schröder, and Stefan Katzenbeisser. Physically uncloneable functions in the universal composition framework. In Phillip Rogaway, editor, *Advances in Cryptology (CRYPTO)*, volume 6841 of *Lecture Notes in Computer Science (LNCS)*, pages 51–70. Springer, Berlin/Heidelberg, Germany, August 2011.
- [34] Mike Burmester, Tri van Le, and Breno de Medeiros. Provably secure ubiquitous systems: Universally composable RFID authentication protocols. In *Conference on Security and Privacy in Communication Networks (SecureComm) and Workshops*, pages 1–9. IEEE, Washington, DC, USA, August 2006.
- [35] Mike Burmester, Tri Van Le, Breno De Medeiros, and Gene Tsudik. Universally composable RFID identification and authentication protocols. *ACM Transactions on Information and Systems Security*, 12(4), April 2009.
- [36] Laurent Bussard and Walid Bagga. Distance-Bounding proof of knowledge to avoid Real-Time attacks. In Ryoichi Sasaki, Sihan Qing, Eiji Okamoto, and Hiroshi Yoshiura, editors, *Security and Privacy in the Age of Ubiquitous Computing*, volume 181, pages 223–238. Springer US, May 2005.
- [37] Jan Camenisch, Susan Hohenberger, and Anna Lysyanskaya. Compact e-cash advances. In Ronald Cramer, editor, *Advances in Cryptology (EUROCRYPT)*, volume 3494 of *Lecture Notes in Computer Science (LNCS)*, page 566. Springer, May 2005.
- [38] Jan Camenisch and Anna Lysyanskaya. A signature scheme with efficient protocols. In Stelvio Cimato, Giuseppe Persiano, and Clemente Galdi, editors, *Security in Communication Networks (SCN)*, volume 2576 of *Lecture Notes in Computer Science (LNCS)*, pages 268–289. Springer, Berlin/Heidelberg, Germany, September 2003.
- [39] Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In Matt Franklin, editor, *Advances in Cryptology (CRYPTO)*, volume 3152 of *Lecture Notes in Computer Science (LNCS)*, pages 1–6. Springer, Berlin/Heidelberg, Germany, August 2004.
- [40] Sébastien Canard, Iwen Coisel, Jonathan Etrog, and Marc Girault. Privacy-preserving RFID systems: Model and constructions. Cryptology ePrint Archive, Report 2010/405, July 2010.
- [41] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *IEEE Symposium on Foundations of Computer Science*, pages 136–145. IEEE, Washington, DC, USA, October 2001.
- [42] Liqun Chen, Paul Morrissey, and Nigel P Smart. DAA: Fixing the pairing based protocols. Cryptology ePrint Archive, Report 2009/198, May 2009.
- [43] Nicolas T. Courtois, Karsten Nohl, and Sean O’Neil. Algebraic attacks on the crypto-1 stream cipher in MiFare Classic and Oyster Cards. Cryptology ePrint Archive, Report 2008/166, April 2008.
- [44] Cas Cremers, Kasper B. Rasmussen, Benedikt Schmidt, and Srdjan Capkun. Distance hijacking attacks on distance bounding protocols. In *IEEE Symposium on Security and Privacy (S&P)*, pages 113–127. IEEE, May 2012.

-
- [45] Ivan Damgård and Michael Pedersen. RFID security: Tradeoffs between security and efficiency. Cryptology ePrint Archive, Report 2006/234, April 2006.
 - [46] Ivan Damgård and Michael Pedersen. RFID security: Tradeoffs between security and efficiency. In Tal Malkin, editor, *Topics in Cryptology (CT-RSA)*, volume 4964 of *Lecture Notes in Computer Science (LNCS)*, pages 318–332. Springer, Berlin/Heidelberg, Germany, April 2008.
 - [47] IvanBjerre Damgård. Collision free hash functions and public key signature schemes. In David Chaum and Wyn L. Price, editors, *Advances in Cryptology (EUROCRYPT)*, volume 304 of *Lecture Notes in Computer Science (LNCS)*, pages 203–216. Springer, Berlin/Heidelberg, Germany, May 1988.
 - [48] Boris Danev, Thomas S. Heydt-Benjamin, and Srdjan Čapkun. Physical-layer identification of RFID devices. In *USENIX Security Symposium*, pages 199–214. USENIX Association, Berkeley, CA, USA, August 2009.
 - [49] Paolo D’Arco, Alessandra Scafuro, and Ivan Visconti. Revisiting DoS attacks and privacy in RFID-enabled networks. In Shlomi Dolev, editor, *International Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS)*, volume 5804 of *Lecture Notes in Computer Science (LNCS)*, pages 76–87. Springer, July 2009.
 - [50] Paolo D’Arco, Alessandra Scafuro, and Ivan Visconti. Semi-Destructive privacy in DoS-enabled RFID systems. In *Workshop on RFID Security (RFIDSec)*. July 2009.
 - [51] Robert H. Deng, Yingjiu Li, Moti Yung, and Yunlei Zhao. A new framework for RFID privacy. In Dimitris Gritzalis, Bart Preneel, and Marianthi Theoharidou, editors, *European Symposium on Research in Computer Security (ESORICS)*, volume 6345 of *Lecture Notes in Computer Science (LNCS)*, pages 1–18. Springer, Berlin/Heidelberg, Germany, 2010.
 - [52] Yvo Desmedt, Claude Goutier, and Samy Bengio. Special uses and abuses of the Fiat-Shamir passport protocol (extended abstract). In Carl Pomerance, editor, *Advances in Cryptology (CRYPTO)*, volume 293 of *Lecture Notes in Computer Science (LNCS)*, pages 21–39. Springer, Berlin/Heidelberg, Germany, August 1988.
 - [53] Ton Deursen and Saša Radomirović. Insider attacks and privacy of RFID protocols. In Svetla Petkova-Nikova, Andreas Pashalidis, and Günther Pernul, editors, *8th European Conference on Public Key Infrastructures, Services and Applications (EuroPKI)*, volume 7163 of *Lecture Notes in Computer Science (LNCS)*, pages 91–105. Springer, 2012.
 - [54] Srinivas Devadas, Edward Suh, Sid Paral, Richard Sowell, Tom Ziola, and Vivek Khandelwal. Design and implementation of PUF-based unclonable RFID ICs for anti-counterfeiting and security applications. In *IEEE International Conference on RFID*, pages 58–64. IEEE, Washington, DC, USA, April 2008.
 - [55] Tassos Dimitriou. A lightweight RFID protocol to protect against traceability and cloning attacks. In *Conference on Security and Privacy for Emerging Areas in Communications Networks (SecureComm)*, pages 59–66. IEEE, Washington, DC, USA, September 2005.
 - [56] Yevgeniy Dodis, Jonathan Katz, Leonid Reyzin, and Adam Smith. Robust fuzzy extractors and authenticated key agreement from close secrets. In Cynthia Dwork, editor, *Advances in Cryptology (CRYPTO)*, volume 4117 of *Lecture Notes in Computer Science (LNCS)*, pages 232–250. Springer, Berlin/Heidelberg, Germany, August 2006.

- [57] Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology (EUROCRYPT)*, volume 3027 of *Lecture Notes in Computer Science (LNCS)*, pages 523–540. Springer, Berlin/Heidelberg, Germany, May 2004.
- [58] François Durvaux, Benoît Gérard, Stéphanie Kerckhof, François Koeune, and François-Xavier Standaert. Intellectual property protection for integrated systems using soft physical hash functions. In Dong Hoon Lee and Moti Yung, editors, *Information Security Applications*, volume 7690 of *Lecture Notes in Computer Science (LNCS)*, pages 208–225. Springer, August 2012.
- [59] Ilze Eichhorn, Patrick Koeberl, and Vincent van der Leest. Logically reconfigurable PUFs: Memory-based secure key storage. In *ACM Workshop on Scalable Trusted Computing (STC)*, pages 59–64. ACM, New York, NY, USA, October 2011.
- [60] Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, July 1985.
- [61] Marc Fischlin and Cristina Onete. Subtle kinks in distance-bounding: An analysis of prominent protocols. In *ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, pages 195–206, New York, NY, USA, April 2013. ACM.
- [62] Aurélien Francillon, Boris Danev, and Srdjan Čapkun. Relay attacks on passive keyless entry and start systems in modern cars. http://www.isoc.org/isoc/conferences/ndss/11/pdf/2_1.pdf, February 2011.
- [63] Dmitry Frumkin and Adi Shamir. Un-Trusted-HB: Security vulnerabilities of Trusted-HB. In *Workshop on RFID Security (RFIDSec)*. July 2009.
- [64] Flavio D. Garcia, Gerhard Koning Gans, Ruben Muijers, Peter Rossum, Roel Verdult, Ronny W. Schreur, and Bart Jacobs. Dismantling MiFare Classic. In Sushil Jajodia and Javier Lopez, editors, *European Symposium on Research in Computer Security (ESORICS)*, volume 5283 of *Lecture Notes in Computer Science (LNCS)*, pages 97–114. Springer, October 2008.
- [65] Flavio D. Garcia and Peter Rossum. Modeling privacy for off-line RFID systems. In Dieter Gollmann, Jean-Louis Lanet, and Julien Iguchi-Cartigny, editors, *International Conference on Smart Card Research and Advanced Application (CARDIS)*, volume 6035 of *Lecture Notes in Computer Science (LNCS)*, pages 194–208. Springer, Berlin/Heidelberg, Germany, 2010.
- [66] Blaise Gassend. Physical random functions. Master’s thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology (MIT), The Stata Center, 32 Vassar Street, Cambridge, Massachusetts 02139, February 2003.
- [67] Blaise Gassend, Dwaine Clarke, Marten van Dijk, and Srinivas Devadas. Controlled physical random functions. In *Annual Computer Security Applications Conference (ACSAC)*, pages 149–160. IEEE, Washington, DC, USA, December 2002.
- [68] Blaise Gassend, Dwaine Clarke, Marten van Dijk, and Srinivas Devadas. Silicon physical random functions. In *ACM Conference on Computer and Communications Security (CCS)*, pages 148–160. ACM, New York, NY, USA, November 2002.

-
- [69] Henri Gilbert, Matt Robshaw, and Herve Sibert. Active attack against HB^+ : A provably secure lightweight authentication protocol. *IET Electronics Letters*, 41(21):1169–1170, October 2005.
 - [70] Henri Gilbert, Matthew Robshaw, and Yannick Seurin. $HB^\#$: Increasing the security and efficiency of HB^+ . In Nigel Smart, editor, *Advances in Cryptology (EUROCRYPT)*, volume 4965 of *Lecture Notes in Computer Science (LNCS)*, pages 361–378. Springer, Berlin/Heidelberg, Germany, April 2008.
 - [71] Marc Girault, Loic Juniot, and Matthew Robshaw. The feasibility of on-the-tag public key cryptography. In *Workshop on RFID Security (RFIDSec)*. Malaga, Spain, July 2007.
 - [72] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, August 1986.
 - [73] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and Systems Sciences*, 28:270–299, April 1984.
 - [74] Philippe Golle, Markus Jakobsson, Ari Juels, and Paul Syverson. Universal re-encryption for mixnets. In Tatsuaki Okamoto, editor, *Topics in Cryptology (CT-RSA)*, volume 2964 of *Lecture Notes in Computer Science (LNCS)*, pages 163–178. Springer, Berlin/Heidelberg, Germany, February 2004.
 - [75] GS1 AISBL. EPCglobal standards. <http://www.gs1.org/epcglobal/standards>, January 2013.
 - [76] Jorge Guajardo, Sandeep Kumar, Geert-Jan Schrijen, and Pim Tuyls. FPGA intrinsic PUFs and their use for IP protection. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems (CHES)*, volume 4727 of *Lecture Notes in Computer Science (LNCS)*, pages 63–80. Springer, Berlin/Heidelberg, Germany, September 2007.
 - [77] Jorge Guajardo, Sandeep S. Kumar, Geert-Jan Schrijen, and Pim Tuyls. Brand and IP protection with physical unclonable functions. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 3186–3189. IEEE, Washington, DC, USA, May 2008.
 - [78] Jung Hoon Ha, Sang Jae Moon, Jianying Zhou, and Jae Cheol Ha. A new formal proof model for RFID location privacy. In Sushil Jajodia and Javier Lopez, editors, *European Symposium on Research in Computer Security (ESORICS)*, volume 5283 of *Lecture Notes in Computer Science (LNCS)*, pages 267–281. Springer, Berlin/Heidelberg, Germany, October 2008.
 - [79] Ghaith Hammouri, Aykutlu Dana, and Berk Sunar. CDs have fingerprints too. In Christophe Clavier and Kris Gaj, editors, *Cryptographic Hardware and Embedded Systems (CHES)*, volume 5747 of *Lecture Notes in Computer Science (LNCS)*, pages 348–362. Springer, Berlin/Heidelberg, Germany, September 2009.
 - [80] Ghaith Hammouri and Berk Sunar. PUF-HB: A tamper-resilient HB based authentication protocol. In Steven M. Bellovin, Rosario Gennaro, Angelos Keromytis, and Moti Yung, editors, *Applied Cryptography and Network Security (ACNS)*, volume 5037 of *Lecture Notes in Computer Science (LNCS)*, pages 346–365. Springer, Berlin/Heidelberg, Germany, June 2008.

- [81] Gerhard P. Hancke. Practical attacks on proximity identification systems. In *IEEE Symposium on Security and Privacy (S&P)*, pages 6–12. IEEE, May 2006.
- [82] Gerhard P. Hancke and Markus G. Kuhn. An RFID distance bounding protocol. In *Security and Privacy for Emerging Areas in Communications Networks (SecureComm)*, pages 67–73, Washington, DC, USA, September 2005. IEEE.
- [83] Dirk Henrici and Paul Muller. Hash-based enhancement of location privacy for radio-frequency identification devices using varying identifiers. In *Conference on Pervasive Computing and Communications (PerCom)*, pages 149–153. IEEE, Washington, DC, USA, March 2004.
- [84] Jens Hermans, Andreas Pashalidis, Frederik Vercauteren, and Bart Preneel. A new RFID privacy model. In Vijay Atluri and Claudia Diaz, editors, *European Symposium on Research in Computer Security (ESORICS)*, volume 6879 of *Lecture Notes in Computer Science (LNCS)*, pages 568–587. Springer, September 2011.
- [85] Anthony Herrewewe, Stefan Katzenbeisser, Roel Maes, Roel Peeters, Ahmad-Reza Sadeghi, Ingrid Verbauwhede, and Christian Wachsmann. Reverse fuzzy extractors: Enabling lightweight mutual authentication for PUF-enabled RFIDs. In Angelos Keromytis, editor, *Financial Cryptography and Data Security (FC)*, volume 7397 of *Lecture Notes in Computer Science (LNCS)*, pages 374–389. Springer, February 2012.
- [86] Thomas S. Heydt-Benjamin, Hee-Jin Chae, Benessa Defend, and Kevin Fu. Privacy for public transportation. In George Danezis and Philippe Golle, editors, *Workshop on Privacy Enhancing Technologies (PET)*, volume 4258 of *Lecture Notes in Computer Science (LNCS)*, pages 1–19. Springer, Berlin/Heidelberg, Germany, June 2006.
- [87] Daniel Holcomb, Wayne Burleson, and Kevin Fu. Initial SRAM state as a fingerprint and source of true random numbers for RFID tags. In *Workshop on RFID Security (RFIDSec)*. July 2007.
- [88] Daniel Holcomb, Wayne P. Burleson, and Kevin Fu. Power-up SRAM state as an identifying fingerprint and source of true random numbers. *IEEE Transactions on Computers*, 58(9):1198–1210, September 2009.
- [89] Daniel E. Holcomb, Amir Rahmati, Mastooreh Salajegheh, Wayne P. Burleson, and Kevin Fu. DRV-fingerprinting: Using data retention voltage of SRAM cells for chip identification. In Jaap-Henk Hoepman and Ingrid Verbauwhede, editors, *Workshop on RFID Security (RFIDSec)*, volume 7739 of *Lecture Notes in Computer Science (LNCS)*, pages 165–179. Springer, July 2013.
- [90] Nicholas J. Hopper and Manuel Blum. Secure human identification protocols. In Colin Boyd, editor, *Advances in Cryptology (ASIACRYPT)*, volume 2248 of *Lecture Notes in Computer Science (LNCS)*, pages 52–66. Springer, Berlin/Heidelberg, Germany, November 2001.
- [91] Michael Hutter, Jörn-Marc Schmidt, and Thomas Plos. RFID and its vulnerability to faults. In Elisabeth Oswald and Pankaj Rohatgi, editors, *Cryptographic Hardware and Embedded Systems (CHES)*, volume 5154 of *Lecture Notes in Computer Science (LNCS)*, pages 363–379. Springer, August 2008.

-
- [92] Tanya Ignatenko, Geert-Jan Schrijen, Boris Škorić, Pim Tuyls, and Frans Willems. Estimating the secrecy-rate of physical unclonable functions with the context-tree weighting method. In *IEEE International Symposium on Information Theory (ISIT)*, pages 499–503. IEEE, Washington, DC, USA, July 2006.
 - [93] International Civil Aviation Organization (ICAO). Machine readable travel documents, doc 9303, August 2006.
 - [94] Intrinsic ID. Product webpage. <http://www.intrinsic-id.com/products.htm>, January 2013.
 - [95] Ari Juels. Minimalist cryptography for low-cost RFID tags (extended abstract). In Carlo Blundo and Stelvio Cimato, editors, *Security in Communication Networks (SCN)*, volume 3352 of *Lecture Notes in Computer Science (LNCS)*, pages 149–164. Springer, Berlin/Heidelberg, Germany, September 2005.
 - [96] Ari Juels. RFID security and privacy: A research survey. *IEEE Journal on Selected Areas in Communications (SAC)*, 24(2):381–394, February 2006.
 - [97] Ari Juels and Ravikanth Pappu. Squealing Euros: Privacy protection in RFID-enabled banknotes. In Rebecca N. Wright, editor, *Financial Cryptography and Data Security (FC)*, volume 2742 of *Lecture Notes in Computer Science (LNCS)*, pages 103–121. Springer, Berlin/Heidelberg, Germany, January 2003.
 - [98] Ari Juels and Stephen A. Weis. Authenticating pervasive devices with human protocols. In Victor Shoup, editor, *Advances in Cryptology (CRYPTO)*, volume 3621 of *Lecture Notes in Computer Science (LNCS)*, pages 293–308. Springer, Berlin/Heidelberg, Germany, November 2005.
 - [99] Ari Juels and Stephen A. Weis. Defining strong privacy for RFID. In *Conference on Pervasive Computing and Communications (PerCom)*, pages 342–347. IEEE, Washington, DC, USA, March 2007.
 - [100] Ari Juels and Stephen A. Weis. Defining strong privacy for RFID. *ACM Transactions on Information and System Security (TISSEC)*, 13, November 2009.
 - [101] Peter Kampstra. Beanplot: A boxplot alternative for visual comparison of distributions. *Journal of Statistical Software*, 28(1):1–9, October 2008.
 - [102] Deniz Karakoyunlu and Berk Sunar. Differential template attacks on PUF enabled cryptographic devices. In *Workshop on Information Forensics and Security (WIFS)*, pages 1–6. IEEE, Washington, DC, USA, December 2010.
 - [103] Suleyman Kardas, Mehmet S. Kiraz, Muhammed A. Bingol, and Huseyin Demirci. A novel RFID distance bounding protocol based on physically unclonable functions. In *Radio Frequency Identification: Security and Privacy Issues (RFIDSec)*, volume 7055 of *Lecture Notes in Computer Science (LNCS)*, pages 78–93. Springer, Berlin/Heidelberg, Germany, June 2011.
 - [104] Timo Kasper, David Oswald, and Christof Paar. New methods for cost-effective side-channel attacks on cryptographic RFIDs. In *Workshop on RFID Security (RFIDSec)*. June 2009.

- [105] Jonathan Katz and Ji S. Shin. Parallel and concurrent security of the HB and HB⁺ protocols. In Serge Vaudenay, editor, *Advances in Cryptology (EUROCRYPT)*, volume 4004 of *Lecture Notes in Computer Science (LNCS)*, pages 73–87. Springer, Berlin/Heidelberg, Germany, June 2006.
- [106] Jonathan Katz and Adam Smith. Analyzing the HB and HB⁺ protocols in the “large error” case. Cryptology ePrint Archive, Report 2006/326, September 2006.
- [107] Stefan Katzenbeisser, Ünal Kocabaş, Vladimir Rožić, Ahmad-Reza Sadeghi, Ingrid Verbauwhede, and Christian Wachsmann. PUFs: Myth, fact or busted? A security evaluation of physically unclonable functions (PUFs) cast in silicon. In Emmanuel Prouff and Patrick Schaumont, editors, *Cryptographic Hardware and Embedded Systems (CHES)*, volume 7428 of *Lecture Notes in Computer Science (LNCS)*, pages 283–301. Springer, September 2012.
- [108] Stefan Katzenbeisser, Ünal Kocabaş, Vincent van der Leest, Ahmad-Reza Sadeghi, Geert-Jan Schrijen, Heike Schröder, and Christian Wachsmann. Recyclable PUFs: Logically reconfigurable PUFs. In *Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, volume 6917, pages 374–389. Springer, Berlin/Heidelberg, Germany, September 2011.
- [109] Stefan Katzenbeisser, Ünal Kocabaş, Vincent van der Leest, Ahmad-Reza Sadeghi, Geert-Jan Schrijen, and Christian Wachsmann. Recyclable PUFs: Logically reconfigurable PUFs. *Journal of Cryptographic Engineering*, 1(3):177–186, September 2011.
- [110] Chong H. Kim, Gildas Avoine, François Koeune, François-Xavier Standaert, and Olivier Pereira. The Swiss-Knife RFID distance bounding protocol. In Pil J. Lee and Jung H. Cheon, editors, *Information Security and Cryptology (ICISC)*, volume 5461 of *Lecture Notes in Computer Science (LNCS)*, pages 98–115. Springer, Berlin/Heidelberg, Germany, December 2009.
- [111] Ilan Kirschenbaum and Avishai Wool. How to build a low-cost, extended-range RFID skimmer. In *USENIX Security Symposium*. USENIX Association, Berkeley, CA, USA, July 2006.
- [112] Hugo Krawczyk. LFSR-based hashing and authentication. In Yvo Desmedt, editor, *Advances in Cryptology (CRYPTO)*, volume 839 of *Lecture Notes in Computer Science (LNCS)*, pages 129–139. Springer, Berlin/Heidelberg, Germany, July 1994.
- [113] Sandeep S. Kumar, Jorge Guajardo, Roel Maes, Geert-Jan Schrijen, and Pim Tuyls. Extended abstract: The butterfly PUF protecting IP on every FPGA. In *Workshop on Hardware-Oriented Security (HOST)*, pages 67–70. IEEE, Washington, DC, USA, June 2008.
- [114] Klaus Kursawe, Ahmad-Reza Sadeghi, Dries Schellekens, Boris Skoric, and Pim Tuyls. Reconfigurable physical unclonable functions — Enabling technology for tamper-resistant storage. In *Workshop on Hardware-Oriented Security and Trust (HOST)*, pages 22–29. IEEE, Washington, DC, USA, July 2009.
- [115] Junzuo Lai, Robert H. Deng, and Yingjiu Li. Revisiting unpredictability-based RFID privacy models. In Jianying Zhou and Moti Yung, editors, *Applied Cryptography and Network Security (ACNS)*, volume 6123 of *Lecture Notes in Computer Science (LNCS)*, pages 475–492. Springer, Berlin/Heidelberg, Germany, June 2010.

-
- [116] Jae W. Lee, Daihyun Lim, Blaise Gassend, Edward G. Suh, Marten van Dijk, and Srinivas Devadas. A technique to build a secret key in integrated circuits for identification and authentication applications. In *Symposium on VLSI Circuits*, pages 176–179. IEEE, Washington, DC, USA, June 2004.
 - [117] Éric Leveil and Pierre-Alain Fouque. An improved LPN algorithm. In Roberto Prisco and Moti Yung, editors, *Security and Cryptography for Networks (SCN)*, volume 4116 of *Lecture Notes in Computer Science (LNCS)*, pages 348–359. Springer, Berlin/Heidelberg, Germany, September 2006.
 - [118] Chae H. Lim and Taekyoung Kwon. Strong and robust RFID authentication enabling perfect ownership transfer. In Peng Ning, Sihan Qing, and Ninghui Li, editors, *International Conference on Information and Communications Security (ICICS)*, volume 4307 of *Lecture Notes in Computer Science (LNCS)*, pages 1–20. Springer, Berlin/Heidelberg, Germany, December 2006.
 - [119] Daihyun Lim. Extracting secret keys from integrated circuits. Master’s thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology (MIT), The Stata Center, 32 Vassar Street, Cambridge, Massachusetts 02139, June 2004.
 - [120] Daihyun Lim, Jae W. Lee, Blaise Gassend, Edward G. Suh, Marten van Dijk, and Srinivas Devadas. Extracting secret keys from integrated circuits. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 13(10):1200–1205, October 2005.
 - [121] Lang Lin, Dan Holcomb, Dilip K. Krishnappa, Prasad Shabadi, and Wayne Burleson. Low-power sub-threshold design of secure physical unclonable functions. In *International Symposium on Low-Power Electronics and Design (ISLPED)*, pages 43–48. IEEE, Washington, DC, USA, August 2010.
 - [122] Joseph K. Liu, Joonsang Baek, Jianying Zhou, Yanjiang Yang, and Jun W Wong. Efficient online/offline identity-based signature for wireless sensor network. *International Journal of Information Security*, 9(4):287–296, August 2010.
 - [123] Changshe Ma, Yingjiu Li, Robert H. Deng, and Tieyan Li. RFID privacy: Relation between two notions, minimal condition, and efficient construction. In *ACM Conference on Computer and Communications Security (CCS)*, pages 54–65, New York, NY, USA, November 2009. ACM.
 - [124] Roel Maes, Pim Tuyls, and Ingrid Verbauwhede. Intrinsic PUFs from flip-flops on reconfigurable devices. In *Benelux Workshop on Information and System Security*. November 2008.
 - [125] Roel Maes and Ingrid Verbauwhede. Physically unclonable functions: A study on the state of the art and future research directions. In Ahmad-Reza Sadeghi and David Naccache, editors, *Towards Hardware-Intrinsic Security*, Information Security and Cryptography, pages 3–37. Springer, Berlin/Heidelberg, Germany, November 2010.
 - [126] Abhranil Maiti, Jeff Casarona, Luke McHale, and Patrick Schaumont. A large scale characterization of RO-PUF. In *Symposium on Hardware-Oriented Security and Trust (HOST)*, pages 94–99. IEEE, Washington, DC, USA, June 2010.

- [127] Mehrdad Majzoobi, Farinaz Koushanfar, and Miodrag Potkonjak. Lightweight secure PUFs. In *International Conference on Computer-Aided Design (ICCAD)*, pages 670–673. IEEE, Washington, DC, USA, November 2008.
- [128] Mehrdad Majzoobi, Farinaz Koushanfar, and Miodrag Potkonjak. Testing techniques for hardware security. In *International Test Conference (ITC)*, pages 1–10. IEEE, Washington, DC, USA, October 2008.
- [129] Mehrdad Majzoobi, Farinaz Koushanfar, and Miodrag Potkonjak. Techniques for design and implementation of secure reconfigurable PUFs. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, 2(1):1–33, March 2009.
- [130] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. Springer, New York, NY, USA, 2007.
- [131] George Marsaglia. The Marsaglia random number CDROM including the diehard battery of tests of randomness. <http://www.stat.fsu.edu/pub/diehard/>, April 2013.
- [132] Maire McLoone and Matthew J. B. Robshaw. New architectures for Low-Cost public key cryptography on RFID tags. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1827–1830. IEEE, May 2007.
- [133] Dominik Merli, Dieter Schuster, Frederic Stumpf, and Georg Sigl. Side-channel analysis of PUFs and fuzzy extractors. In Jonathan M. McCune, Boris Balacheff, Adrian Perrig, Ahmad-Reza Sadeghi, Angela Sasse, and Yolanta Beres, editors, *Trust and Trustworthy Computing (TRUST)*, volume 6740 of *Lecture Notes in Computer Science (LNCS)*, pages 33–47. Springer, Berlin/Heidelberg, Germany, June 2011.
- [134] Mala Mitra. Privacy for RFID systems to prevent tracking and cloning. *International Journal of Computer Science and Network Security (IJCSNS)*, 8(1), January 2008.
- [135] David Molnar and David Wagner. Privacy and security in library RFID: Issues, practices and architectures. In *ACM Conference on Computer and Communications Security (CCS)*, pages 210–219. ACM, New York, NY, USA, October 2004.
- [136] NFC Forum. NFC Forum specifications. <http://www.nfc-forum.org/specs/>, January 2013.
- [137] Ching Y. Ng, Willy Susilo, Yi Mu, and Rei Safavi-Naini. RFID privacy models revisited. In Sushil Jajodia and Javier Lopez, editors, *European Symposium on Research in Computer Security (ESORICS)*, volume 5283 of *Lecture Notes in Computer Science (LNCS)*, pages 251–266. Springer, Berlin/Heidelberg, Germany, October 2008.
- [138] Ching Y. Ng, Willy Susilo, Yi Mu, and Rei Safavi-Naini. New privacy results on synchronized RFID authentication protocols against tag tracing. In Michael Backes and Peng Ning, editors, *European Symposium on Research in Computer Security (ESORICS)*, volume 5789 of *Lecture Notes in Computer Science (LNCS)*, pages 321–336. Springer, September 2009.
- [139] Rishab Nithyanand, Gene Tsudik, and Ersin Uzun. Readers behaving badly. In Dimitris Gritzalis, Bart Preneel, and Marianthi Theoharidou, editors, *European Symposium on Research in Computer Security (ESORICS)*, volume 6345 of *Lecture Notes in Computer Science (LNCS)*, pages 19–36. Springer, Berlin/Heidelberg, Germany, September 2010.

-
- [140] Karsten Nohl and Henryk Plötz. MiFare — Little security despite obscurity. <http://events.ccc.de/congress/2007/Fahrplan/events/2378.en.html>, December 2007.
- [141] NXP Semiconductors. MiFare smartcard IC's. <http://mifare.net/products/mifare-smartcard-ic-s/>, January 2013.
- [142] NXP Semiconductors. SmartMX: High security microcontroller IC. http://www.mifare.net/files/3013/0079/2103/SmartMX%20Leaflet_Oct10.pdf, November 2013.
- [143] Miyako Ohkubo, Koutarou Suzuki, and Shingo Kinoshita. Cryptographic approach to “privacy-friendly” tags, November 2003.
- [144] Miyako Ohkubo, Koutarou Suzuki, and Shingo Kinoshita. Efficient hash-chain based RFID privacy protection scheme. In *International Conference on Ubiquitous Computing (Ubi-comp), Workshop Privacy: Current Status and Future Directions*. September 2004.
- [145] Yossef Oren, Ahmad-Reza Sadeghi, and Christian Wachsmann. On the effectiveness of the remanence decay side-channel to clone memory-based PUFs. In *Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, volume 8086 of *Lecture Notes in Computer Science (LNCS)*, pages 107–125. Springer, Berlin/Heidelberg, Germany, August 2013.
- [146] Khaled Ouafi and Raphael C. W Phan. Privacy of recent RFID authentication protocols. In Liqun Chen, Yi Mu, and Willy Susilo, editors, *Information Security Practice and Experience (ISPEC)*, volume 4991 of *Lecture Notes in Computer Science (LNCS)*, pages 263–277. Springer, Berlin/Heidelberg, Germany, April 2008.
- [147] Khaled Ouafi and RaphaelC Phan. Traceable privacy of recent provably-secure RFID protocols. In Steven M. Bellovin, Rosario Gennaro, Angelos Keromytis, and Moti Yung, editors, *Applied Cryptography and Network Security (ACNS)*, volume 5037 of *Lecture Notes in Computer Science (LNCS)*, pages 479–489, Berlin/Heidelberg, Germany, June 2008. Springer.
- [148] Erdiñç Öztürk, Ghaith Hammouri, and Berk Sunar. Towards robust low cost authentication for pervasive devices. In *Conference on Pervasive Computing and Communications (PerCom)*, pages 170–178. IEEE, Washington, DC, USA, March 2008.
- [149] Pascal Paillier. Public-Key cryptosystems based on composite degree residuosity classes. In Jacques Stern, editor, *Advances in Cryptology (EUROCRYPT)*, volume 1592 of *Lecture Notes in Computer Science (LNCS)*, pages 223–238. Springer, Berlin/Heidelberg, Germany, May 1999.
- [150] Radu I. Paise and Serge Vaudenay. Mutual authentication in RFID: Security and privacy. In *ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, pages 292–299. ACM, New York, NY, USA, March 2008.
- [151] Ravikanth Pappu, Ben Recht, Jason Taylor, and Neil Gershenfeld. Physical one-way functions. *Science*, 297(5589):2026–2030, September 2002.
- [152] Roel Peeters and Jens Hermans. Wide strong private RFID identification based on zero-knowledge. Cryptology ePrint Archive, Report 2012/389, July 2012.

- [153] Pedro Peris-Lopez, Julio C. Hernandez-Castro, Juan M. Estevez-Tapiador, and Arturo Ribagorda. RFID systems: A survey on security threats and proposed solutions. In Pedro Cuenca and Luiz Orozco-Barbosa, editors, *Personal Wireless Communications (PCW)*, volume 4217 of *Lecture Notes in Computer Science (LNCS)*, pages 159–170. Springer, Berlin/Heidelberg, Germany, September 2006.
- [154] Pedro Peris-Lopez, Julio C. Hernandez-Castro, Juan M. E. Tapiador, Esther Palomar, and Jan C. A. van der Lubbe. Cryptographic puzzles and distance-bounding protocols: Practical tools for RFID security. In *IEEE International Conference on RFID*, pages 45–52. IEEE, April 2010.
- [155] Krzysztof Pietrzak. A Leakage-Resilient mode of operation. In Antoine Joux, editor, *Advances in Cryptology (EUROCRYPT)*, volume 5479 of *Lecture Notes in Computer Science (LNCS)*, pages 462–482. Springer, April 2009.
- [156] Damith C. Ranasinghe, Daniel W. Engels, and Peter H Cole. Security and privacy: Modest proposals for low-cost RFID systems. In *Auto-ID Labs Research Workshop*. September 2004.
- [157] Kasper B. Rasmussen and Srdjan Čapkun. Realization of RF distance bounding. In *USENIX Conference on Security*, page 25, Berkeley, CA, USA, December 2010. USENIX Association.
- [158] Henry P. Romero, Kate A. Remley, Dylan F. Williams, and Chih-Ming Wang. Electromagnetic measurements for counterfeit detection of radio frequency identification cards. *IEEE Transactions on Microwave Theory and Techniques*, 57(5):1383–1387, May 2009.
- [159] Henry P. Romero, Kate A. Remley, Dylan F. Williams, Chih-Ming Wang, and Timothy X. Brown. Identifying RF identification cards from measurements of resonance and carrier harmonics. *IEEE Transactions on Microwave Theory and Techniques*, 58(7):1758–1765, July 2010.
- [160] Ulrich Rührmair, Frank Sehnke, Jan Sölter, Gideon Dror, Srinivas Devadas, and Jürgen Schmidhuber. Modeling attacks on physical unclonable functions. In *ACM Conference on Computer and Communications Security (CCS)*, pages 237–249. ACM, New York, NY, USA, October 2010.
- [161] Ulrich Rührmair, Jan Sölter, and Frank Sehnke. On the foundations of physical unclonable functions. Cryptology ePrint Archive, Report 2009/277, June 2009.
- [162] Andrew Rukhin, Juan Soto, James Nechvatal, Miles Smid, Elaine Barker, Stefan Leigh, Mark Levenson, Mark Vangel, David Banks, Alan Heckert, James Dray, and San Vo. A statistical test suite for random and pseudorandom number generators for cryptographic applications. Special Publication 800-22 Revision 1a, NIST, April 2010.
- [163] Ahmad-Reza Sadeghi, Ivan Visconti, and Christian Wachsmann. User privacy in transport systems based on RFID e-tickets. In *Workshop on Privacy in Location-Based Applications (PiLBA)*. October 2008.
- [164] Ahmad-Reza Sadeghi, Ivan Visconti, and Christian Wachsmann. Anonymizer-enabled security and privacy for RFID. In Juan Garay, Atsuko Miyaji, and Akira Otsuka, editors, *Cryptology and Network Security (CANS)*, volume 5888 of *Lecture Notes in Computer Science (LNCS)*, pages 134–153. Springer, Berlin/Heidelberg, Germany, December 2009.

-
- [165] Ahmad-Reza Sadeghi, Ivan Visconti, and Christian Wachsmann. Efficient RFID security and privacy with anonymizers. In *Workshop on RFID Security (RFIDSec)*. July 2009.
 - [166] Ahmad-Reza Sadeghi, Ivan Visconti, and Christian Wachsmann. Location privacy in RFID applications. In Claudio Bettini, Sushil Jajodia, Pierangela Samarati, and X. Sean Wang, editors, *Privacy in Location-Based Applications*, volume 5599 of *Lecture Notes in Computer Science (LNCS)*, pages 127–150. Springer, Berlin/Heidelberg, Germany, August 2009.
 - [167] Ahmad-Reza Sadeghi, Ivan Visconti, and Christian Wachsmann. Enhancing RFID security and privacy by physically unclonable functions. In Ahmad-Reza Sadeghi and David Naccache, editors, *Towards Hardware-Intrinsic Security*, Information Security and Cryptography, pages 281–305. Springer, Berlin/Heidelberg, Germany, November 2010.
 - [168] Ahmad-Reza Sadeghi, Ivan Visconti, and Christian Wachsmann. PUF-enhanced RFID security and privacy. In *Secure Component and System Identification (SECSI)*. April 2010.
 - [169] Junichiro Saito, Jae-Cheol Ryou, and Kouichi Sakurai. Enhancing privacy of universal re-encryption scheme for RFID tags. In Laurence T. Yang, Minyi Guo, Guang R. Gao, and Niraj K. Jha, editors, *International Conference on Embedded and Ubiquitous Computing (EUC)*, volume 3207 of *Lecture Notes in Computer Science (LNCS)*, pages 55–84. Springer, Berlin/Heidelberg, Germany, August 2004.
 - [170] Nitesh Saxena and Jonathan Voris. We can remember it for you wholesale: Implications of data remanence on the use of RAM for true random number generation on RFID tags (RFIDSec 2009). July 2009.
 - [171] Steffen Schulz, Ahmad-Reza Sadeghi, and Christian Wachsmann. Short paper: Lightweight remote attestation using physical functions. In *ACM Conference on Wireless Network Security (WiSec)*, pages 109–114. ACM, New York, NY, USA, June 2011.
 - [172] Georgios Selimis, Mario Konijnenburg, Maryam Ashouei, Jos Huisken, Harmke de Groot, Vicnent van der Leest, Geert-Jan Schrijen, M. van Hulst, and P. Tuyls. Evaluation of 90nm 6T-SRAM as physical unclonable function for secure key generation in wireless sensor nodes. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 567–570. IEEE, Washington, DC, USA, May 2011.
 - [173] Saloomesh Shariati, François Koeune, and François-Xavier Standaert. Security analysis of image-based PUFs for anti-counterfeiting. In Bart Decker and David W. Chadwick, editors, *Communications and Multimedia Security*, volume 7394 of *Lecture Notes in Computer Science (LNCS)*, pages 26–38. Springer, September 2012.
 - [174] Saloomesh Shariati, François-Xavier Standaert, Laurent Jacques, and Benoit Macq. Analysis and experimental evaluation of image-based PUFs. 2(3):189–206, October 2012.
 - [175] Dave Singelée and Bart Preneel. Distance bounding in noisy environments. In Frank Stajano, Catherine Meadows, Srdjan Capkun, and Tyler Moore, editors, *Security and Privacy in Ad-hoc and Sensor Networks*, volume 4572 of *Lecture Notes in Computer Science (LNCS)*, pages 101–115. Springer, Berlin/Heidelberg, Germany, July 2007.
 - [176] Boyeon Song and Chris J. Mitchell. RFID authentication protocol for low-cost tags. In *ACM Conference on Wireless Network Security (WiSec)*, pages 140–147. ACM, New York, NY, USA, March 2008.

- [177] Sony. FeliCa website. <http://www.sony.net/Products/felica/>, January 2013.
- [178] Spirtech. CALYPSO functional specification: Card application, version 1.3. <http://calypso.spirtech.net/>, January 2013.
- [179] Ying Su, Jeremy Holleman, and Brian P Otis. A digital 1.6 pJ/bit chip identification circuit using process variations. *IEEE Journal of Solid-State Circuits*, 43(1):69–77, January 2008.
- [180] Edward G. Suh and Srinivas Devadas. Physical unclonable functions for device authentication and secret key generation. In *ACM/IEEE Design Automation Conference (DAC)*, pages 9–14. IEEE, Washington, DC, USA, June 2007.
- [181] The Economist. Security technology: Where’s the smart money?, February 2002.
- [182] Carlos Tokunaga, David T. Blaauw, and Trevor N. Mudge. True random number generator with a metastability-based quality control. In *IEEE International Solid-State Circuits Conference (ISSCC)*, pages 404–611. IEEE, Washington, DC, USA, February 2007.
- [183] Gene Tsudik. YA-TRAP: Yet another trivial RFID authentication protocol. In *Conference on Pervasive Computing and Communications (PerCom)*, pages 640–643. IEEE, Washington, DC, USA, March 2006.
- [184] Pim Tuyls and Lejla Batina. RFID-tags for anti-counterfeiting. In David Pointcheval, editor, *Topics in Cryptology (CT-RSA)*, volume 3860 of *Lecture Notes in Computer Science (LNCS)*, pages 115–131. Springer, Berlin/Heidelberg, Germany, February 2006.
- [185] Pim Tuyls, Geert-Jan Schrijen, Boris Škorić, Jan van Geloven, Nynke Verhaegh, and Rob Wolters. Read-proof hardware from protective coatings. In Louis Goubin and Mitsuru Matsui, editors, *Cryptographic Hardware and Embedded Systems (CHES)*, volume 4249 of *Lecture Notes in Computer Science (LNCS)*, pages 369–383. Springer, Berlin/Heidelberg, Germany, October 2006.
- [186] Pim Tuyls, Boris Škorić, Tanya Ignatenko, Frans Willems, and Geert-Jan Schrijen. Entropy estimation for optical PUFs based on context-tree weighting methods. In Pim Tuyls, Boris Škorić, and Tom Kevenaar, editors, *Security with Noisy Data*, pages 217–233. Springer, London, UK, October 2007.
- [187] Pim Tuyls, Boris Škorić, Sjoerd Stallinga, Anton H. M. Akkermans, and Wil Ophey. Information-theoretic security analysis of physical uncloneable functions. In Andrew Patrick and Moti Yung, editors, *Financial Cryptography and Data Security (FC)*, volume 3570 of *Lecture Notes in Computer Science (LNCS)*, page 578. Springer, Berlin/Heidelberg, Germany, February 2005.
- [188] Vincent van der Leest, Geert-Jan Schrijen, Helena Handschuh, and Pim Tuyls. Hardware intrinsic security from D flip-flops. In *ACM Workshop on Scalable Trusted Computing (STC)*, pages 53–62. ACM, New York, NY, USA, October 2010.
- [189] Ton van Deursen and Saša Radomirović. On a new formal proof model for RFID location privacy. *Information Processing Letters*, 110(2):57–61, December 2009.
- [190] Serge Vaudenay. On privacy models for RFID. In Kaoru Kurosawa, editor, *Advances in Cryptology (ASIACRYPT)*, volume 4833 of *Lecture Notes in Computer Science (LNCS)*, pages 68–87. Springer, Berlin/Heidelberg, Germany, December 2007.

-
- [191] Serge Vaudenay. Privacy models for RFID schemes. In Siddika Berna Ors Yalcin, editor, *Radio Frequency Identification: Security and Privacy Issues (RFIDSec)*, volume 6370 of *Lecture Notes in Computer Science (LNCS)*, page 65. Springer, June 2010.
 - [192] Verayo Inc. Product webpage. <http://www.verayo.com/product/products.html>, January 2013.
 - [193] Boris Škorić, Stefan Maubach, Tom Kevenaar, and Pim Tuyls. Information-theoretic analysis of capacitive physical unclonable functions. *Journal of Applied Physics*, 100(2):024902–024902–11, July 2006.
 - [194] Boris Škorić, Stefan Maubach, Tom Kevenaar, and Pim Tuyls. Information-theoretic analysis of coating PUFs. Cryptology ePrint Archive, Report 2006/101, March 2006.
 - [195] Boris Škorić, Pim Tuyls, and Wil Ophey. Robust key extraction from physical uncloneable functions. In John Ioannidis, Angelos Keromytis, and Moti Yung, editors, *Applied Cryptography and Network Security (ACNS)*, volume 3531 of *Lecture Notes in Computer Science (LNCS)*, pages 99–135. Springer, Berlin/Heidelberg, Germany, June 2005.
 - [196] Stephen A. Weis, Sanjay E. Sarma, Ronald L. Rivest, and Daniel W. Engels. Security and privacy aspects of low-cost radio frequency identification systems. In Dieter Hutter, Günter Müller, Werner Stephan, and Markus Ullmann, editors, *Conference on Pervasive Computing and Communications (PerCom)*, volume 2802 of *Lecture Notes in Computer Science (LNCS)*, pages 50–59. Springer, Berlin/Heidelberg, Germany, March 2004.
 - [197] Frans Willems. CTW website. <http://www.ele.tue.nl/ctw/>.
 - [198] Frans M. J. Willems, Yuri M. Shtarkov, and Tjalling J Tjalkens. The context-tree weighting method: Basic properties. *IEEE Transactions on Information Theory*, 41(3):653–664, May 1995.
 - [199] Yu Yao, Jiawei Huang, Sudhanshu Khanna, Abhi Shelat, Benton Highsmith Calhoun, John Lach, and David Evans. A sub-0.5V lattice-based public-key encryption scheme for RFID platforms in 130nm CMOS. In *Workshop on RFID Security (RFIDSec Asia)*, volume 6 of *Cryptology and Information Security*, pages 96–113. IOS Press, Wuxi, China, April 2011.
 - [200] Davide Zanetti, Boris Danev, and Srdjan Čapkun. Physical-layer identification of UHF RFID tags. In *Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 353–364, New York, NY, USA, 2010. ACM.
 - [201] Davide Zanetti, Pascal Sachs, and Srdjan Capkun. On the practicality of UHF RFID fingerprinting: How real is the RFID tracking problem? In Simone Fischer-Hübner and Nicholas Hopper, editors, *Privacy Enhancing Technologies (PET)*, volume 6794 of *Lecture Notes in Computer Science (LNCS)*, pages 97–116. Springer, Berlin/Heidelberg, Germany, 2011.

About the Author

NAME:	Christian Wachsmann
--------------	----------------------------

CONTACT:	ch.wachsmann@gmail.com
-----------------	------------------------

PERSONAL:	Born on April 26, 1981 in Coburg, Germany. Citizen of Germany.
------------------	--

EDUCATION:	
Since 2011	Doctoral Candidate at Technische Universität Darmstadt, Germany. Advisors: Prof. Dr.-Ing. Ahmad-Reza Sadeghi (Technische Universität Darmstadt, Germany) and Prof. Dr. Ir. Bart Preneel (Katholieke Universiteit Leuven, Belgium).
2008 – 2010	Ph.D. Student at Ruhr-Universität Bochum, Germany. Advisor: Prof. Dr.-Ing. Ahmad-Reza Sadeghi (Ruhr-Universität Bochum, Germany).
Aug. 2008	Diploma in IT Security (Dipl.-Ing. (Univ.) equivalent to M.Sc.) at Ruhr-Universität Bochum, Germany. Thesis: <i>Privacy-Enhancing Cryptographic Systems for RFID-based E-Tickets</i> . Advisors: Prof. Dr.-Ing. Ahmad-Reza Sadeghi (Ruhr-Universität Bochum, Germany) and Prof. Dr. Giuseppe Persiano (Università di Salerno, Italy).
2002 – 2008	Student of IT-Security at Ruhr-Universität Bochum, Germany. Courses on cryptography, system and network security, computer architectures, computer networks, software engineering, computer engineering, information technology, signal processing and transmission and electrical engineering.
Jul. 2002	Baccalaureate (Abitur) at Gymnasium Ernestinum Coburg, Germany.

WORK EXPERIENCE:	
Since 2011	Research Assistant at System Security Lab, Technische Universität Darmstadt and Center for Advanced Security Research Darmstadt (CASED), Germany. Main project: Foundations for Forgery-Resistant Security Hardware (UNIQUE).
2008 – 2010	Research Assistant at System Security Lab, Horst Görtz Institute for IT Security (HGI), Ruhr-Universität Bochum, Germany. Main projects: Foundations for Forgery-Resistant Security Hardware (UNIQUE), Signal Processing in the Encrypted Domain (SPEED), European Network of Excellence in Cryptology (ECRYPT).
2008 – 2011	Mentor at International School of IT Security (ISITS), Bochum, Germany for the course <i>Secure Systems and Protocols</i> .
2004 – 2010	Teaching Assistant at System Security Lab, Horst Görtz Institute for IT Security (HGI), Ruhr-Universität Bochum, Germany for the courses <i>System Security</i> (2004 – 2010) and <i>Trusted Computing</i> (2010) held by Prof. Dr.-Ing. Ahmad-Reza Sadeghi.
2004 – 2008	Student Assistant at System Security Lab, Horst Görtz Institute for IT Security (HGI), Ruhr-Universität Bochum, Germany. Main project: Signal Processing in the Encrypted Domain (SPEED).

HONORS:	Member of Ruhr-University Bochum Research School (part of German Excellence Initiative), 2008 – 2010. Eduard-Rhein Youth Award (Eduard-Rhein-Jugendpreis), 2000.
----------------	---

Peer-Reviewed Publications

- [1] Frederik Armknecht, Liqun Chen, Ahmad-Reza Sadeghi, and Christian Wachsmann. Anonymous authentication for RFID systems. In Siddika Ors Yalcin, editor, *Radio Frequency Identification: Security and Privacy Issues (RFIDSec)*, volume 6370 of *Lecture Notes in Computer Science (LNCS)*, pages 158–175. Springer, Berlin/Heidelberg, Germany, June 2010.
- [2] Frederik Armknecht, Roel Maes, Ahmad-Reza Sadeghi, François-Xavier Standaert, and Christian Wachsmann. A formal foundation for the security features of physical functions. In *IEEE Symposium on Security and Privacy (S&P)*, pages 397–412. IEEE, Washington, DC, USA, May 2011.
- [3] Frederik Armknecht, Ahmad-Reza Sadeghi, Alessandra Scafuro, Ivan Visconti, and Christian Wachsmann. Impossibility results for RFID privacy. In Marina Gavrilova, C. J. Kenneth Tan, and Edward Moreno, editors, *Transactions on Computational Science XI*, volume 6480 of *Lecture Notes in Computer Science (LNCS)*, pages 39–63. Springer, Berlin/Heidelberg, Germany, December 2010.
- [4] Frederik Armknecht, Ahmad-Reza Sadeghi, Steffen Schulz, and Christian Wachsmann. A security framework for the analysis and design of software attestation. In *ACM Conference on Computer and Communications Security (CCS)*, pages 1–12. ACM, New York, NY, USA, November 2013.
- [5] Frederik Armknecht, Ahmad-Reza Sadeghi, Ivan Visconti, and Christian Wachsmann. On RFID privacy with mutual authentication and tag corruption. In Jianying Zhou and Moti Yung, editors, *Applied Cryptography and Network Security (ACNS)*, volume 6123 of *Lecture Notes in Computer Science (LNCS)*, pages 493–510. Springer, Berlin/Heidelberg, Germany, June 2010.
- [6] Christoph Busold, Ahmed Taha, Christian Wachsmann, Alexandra Dmitrienko, Hervé Seudié, Majid Sobhani, and Ahmad-Reza Sadeghi. Smart keys for cyber-cars: Secure smartphone-based NFC-enabled car immobilizer. In *ACM Conference on Data and Application Security and Privacy (CODASPY)*, pages 233–242. ACM, New York, NY, USA, February 2013.
- [7] Liqun Chen, Kurt Dietrich, Hans Löhr, Ahmad-Reza Sadeghi, Christian Wachsmann, and Johannes Winter. Lightweight anonymous authentication with TLS and DAA for embedded mobile devices. In Mike Burmester, Gene Tsudik, Spyros Magliveras, and Ivana Ilic, editors, *Information Security Conference (ISC)*, volume 6531 of *Lecture Notes in Computer Science (LNCS)*, pages 84–98. Springer, Berlin/Heidelberg, Germany, October 2011.
- [8] Alexandra Dmitrienko, Ahmad-Reza Sadeghi, Sandeep Tamrakar, and Christian Wachsmann. SmartTokens: Delegable access control with NFC-enabled smartphones. In Stefan Katzenbeisser, Edgar Weippl, L. Jean Camp, Melanie Volkamer, Mike Reiter, and Xinwen Zhang, editors, *Trust and Trustworthy Computing (TRUST)*, volume 7344 of *Lecture Notes in Computer Science (LNCS)*, pages 219–238. Springer, Berlin/Heidelberg, Germany, June 2012.
- [9] Anthony Herrewé, Stefan Katzenbeisser, Roel Maes, Roel Peeters, Ahmad-Reza Sadeghi, Ingrid Verbauwhede, and Christian Wachsmann. Reverse fuzzy extractors: Enabling lightweight mutual authentication for PUF-enabled RFIDs. In Angelos Keromytis, editor, *Financial Cryptography and Data Security (FC)*, volume 7397 of *Lecture Notes in Computer Science (LNCS)*, pages 374–389. Springer, February 2012.
- [10] Stefan Katzenbeisser, Ünal Kocabaş, Vladimir Rožić, Ahmad-Reza Sadeghi, Ingrid Verbauwhede, and Christian Wachsmann. PUFs: Myth, fact or busted? A security evaluation of physically unclonable functions (PUFs) cast in silicon. In Emmanuel Prouff and Patrick Schaumont, editors, *Cryptographic Hardware and Embedded Systems (CHES)*, volume 7428 of *Lecture Notes in Computer Science (LNCS)*, pages 283–301. Springer, September 2012.
- [11] Stefan Katzenbeisser, Ünal Kocabaş, Vincent van der Leest, Ahmad-Reza Sadeghi, Geert-Jan Schrijen, Heike Schröder, and Christian Wachsmann. Recyclable PUFs: Logically reconfigurable

- PUFs. In *Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, volume 6917, pages 374–389. Springer, Berlin/Heidelberg, Germany, September 2011.
- [12] Joonho Kong, Farinaz Koushanfar, Ahmad-Reza Sadeghi, and Christian Wachsmann. PUFatt: Embedded platform attestation based on novel processor-based PUFs. In *Design Automation Conference (DAC)*. ACM, New York, NY, USA, June 2014. To appear.
 - [13] Yossef Oren, Ahmad-Reza Sadeghi, and Christian Wachsmann. On the effectiveness of the remanence decay side-channel to clone memory-based PUFs. In *Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, volume 8086 of *Lecture Notes in Computer Science (LNCS)*, pages 107–125. Springer, Berlin/Heidelberg, Germany, August 2013.
 - [14] Ahmad-Reza Sadeghi, Marcel Selhorst, Christian Stübke, Christian Wachsmann, and Marcel Winandy. TCG inside? A note on TPM specification compliance. In *ACM Workshop on Scalable Trusted Computing (STC)*, pages 47–56. ACM, New York, NY, USA, November 2006.
 - [15] Ahmad-Reza Sadeghi, Ivan Visconti, and Christian Wachsmann. Anonymizer-enabled security and privacy for RFID. In Juan Garay, Atsuko Miyaji, and Akira Otsuka, editors, *Cryptology and Network Security (CANS)*, volume 5888 of *Lecture Notes in Computer Science (LNCS)*, pages 134–153. Springer, Berlin/Heidelberg, Germany, December 2009.
 - [16] Steffen Schulz, Ahmad-Reza Sadeghi, and Christian Wachsmann. Short paper: Lightweight remote attestation using physical functions. In *ACM Conference on Wireless Network Security (WiSec)*, pages 109–114. ACM, New York, NY, USA, June 2011.

Other Publications

- [17] Stefan Katzenbeisser, Ünal Kocabaş, Vincent van der Leest, Ahmad-Reza Sadeghi, Geert-Jan Schrijen, and Christian Wachsmann. Recyclable PUFs: Logically reconfigurable PUFs. *Journal of Cryptographic Engineering*, 1(3):177–186, September 2011.
- [18] Ünal Kocabaş, Ahmad-Reza Sadeghi, Christian Wachsmann, and Steffen Schulz. Poster: Practical embedded remote attestation using physically unclonable functions. In *ACM Conference on Computer and Communications Security (CCS)*, pages 797–800. ACM, New York, NY, USA, October 2011.
- [19] Ahmad-Reza Sadeghi, Steffen Schulz, and Christian Wachsmann. Physical security primitives: A survey on physically unclonable functions (PUFs) and PUF-based security solutions. In Konstantinos Markantonakis and Keith Mayes, editors, *Secure Smart Embedded Devices: Platforms and Applications*, pages 429–449. Springer, Berlin/Heidelberg, Germany, 2014.
- [20] Ahmad-Reza Sadeghi, Ivan Visconti, and Christian Wachsmann. Location privacy in RFID applications. In Claudio Bettini, Sushil Jajodia, Pierangela Samarati, and X. Sean Wang, editors, *Privacy in Location-Based Applications*, volume 5599 of *Lecture Notes in Computer Science (LNCS)*, pages 127–150. Springer, Berlin/Heidelberg, Germany, August 2009.
- [21] Ahmad-Reza Sadeghi, Ivan Visconti, and Christian Wachsmann. Enhancing RFID security and privacy by physically unclonable functions. In Ahmad-Reza Sadeghi and David Naccache, editors, *Towards Hardware-Intrinsic Security*, Information Security and Cryptography, pages 281–305. Springer, Berlin/Heidelberg, Germany, November 2010.
- [22] Ahmad-Reza Sadeghi and Christian Wachsmann. Trusted computing. In Burton Rosenberg, editor, *Handbook of Financial Cryptography and Security*, Cryptography and Network Security Series, pages 221–256. Chapman & Hall/CRC, August 2010.
- [23] Ahmad-Reza Sadeghi and Christian Wachsmann. Location Privacy in RFID Systemen. *Digma — Zeitschrift für Datenrecht und Informationssicherheit*, 2011(2):70–75, June 2011.

Erklärung gemäß §9 der Promotionsordnung

Hiermit versichere ich, die vorliegende Dissertation selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel verfasst zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 29. Juli 2013

Dipl.-Ing. Christian Wachsmann